**GLOSSARY FOR THE COURSE “Algorithms and data structures”**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Term | Transcription | Definition | Russian equivalent |
| 1. 1 | Data structure | [ˈdeɪtə ˈstrʌkʧə] | Is the scheme of organizing related information | Структура данных | |
| 1. 2 | Linear data structure | [ˈlɪnɪə ˈdeɪtə ˈstrʌkʧə] | Linear data structure traverses the data elements sequentially, in which only one data element can directly be reached | Линейная структура данных | |
| 1. 3 | Non linear data structure | [ˈlɪnɪə ˈdeɪtə ˈstrʌkʧə] | In the Non linear data structure every data item is attached to several other data items in a way that is specific for reflecting relationships. The data items are not arranged in a sequential structure | Нелинейная структура данных | |
| 1. 4 | Hierarchical structure | [haɪəˈrɑːkɪkl ˈstrʌkʧə] | Is a structure of data having several levels arranged in a treelike structure | Иерархическая структура | |
| 1. 5 | Linear data structures with direct access | [ˈlɪnɪə ˈdeɪtə ˈstrʌkʧə wɪð dɪˈrekt ˈækses] | In the Linear data structures with direct access the computer can calculate exactly where the data has been stored and can go straight to it directly | Линейная структура данных с прямым доступом | |
|  | Linear data structures with serial access | [ˈlɪnɪə ˈdeɪtə ˈstrʌkʧə wɪð ˈsɪərɪəl ˈækses] | In the Linear data structures with serial access data is accessed by starting at the beginning and then searched through, in order/sequence, until the required information is found | Линейная структура данных с последовательным доступом | |
|  | Big O notation (with a capital letter O), also called Landau's symbol | [bɪg əʊ nəʊˈteɪʃn] | Is the symbolism used in computer science to describe the asymptotic behavior of functions. It tells you how fast a function grows or declines. | Нотация «Большое О» | |
|  | Time complexity | [taɪm kəmˈpleksɪtɪ] | Time complexity signifies the total time required by the program to run to completion. | Временная сложность | |
|  | Searching | [ˈsɜːʧɪŋ] | Is the trying to find the information you need. | Поиск | |
|  | Linear search | [ˈlɪnɪə sɜːʧ] | Linear search sequentially moves through your list looking for a matching value | Линейный поиск | |
|  | Binary search | [ˈbaɪnərɪ sɜːʧ] | Binary search inspects an element and compares it with the search argument key.  If it is equal to key, the search terminates;  if it is less than key, we eliminate from further search all elements with index less or equal to mid;  and if it is greater than x, all with index greater or equal to mid can be eliminated. | Бинарный поиск | |
|  | Sorting | [ˈsɔːtɪŋ] | Is the process of rearranging a set of objects in a specific order | Сортировка | |
|  | Internal sorting | [ɪnˈtɜːnl ˈsɔːtɪŋ] | Is any data sorting process that takes place entirely within the main memory of a computer. | Внутренняя сортировка | |
|  | External sorting | [eksˈtɜːnl ˈsɔːtɪŋ] | Is a class of sorting algorithms that can handle massive amounts of data. External sorting is required when the data being sorted do not fit into the main memory of a computing device (usually RAM) and instead they must reside in the slower external memory (usually a hard drive). | Внешняя сортировка | |
|  | Sorting by Insertion | [ˈsɔːtɪŋ baɪ ɪnˈsɜːʃn] | The items (cards) are divided into a sorted sequence a[0] ... a[i-1] and a source sequence a[i] ... a[n-1]. In each step, starting with i = 1, the i th element of the source sequence is picked and transferred into the sorted sequence by inserting it at the appropriate place. | Сортировка вставками | |
|  | Sorting by Selection | [ˈsɔːtɪŋ baɪ sɪˈlekʃn] | This method is based on the following principle:  1. Select the item with the smallest key.  2. Exchange it with the first item a[0].  3. Then repeat these operations with the remaining n-1 items, then with n-2 items, until only one item - - the largest -- is left. | Сортировка выбором | |
|  | Sorting by Exchange | [ˈsɔːtɪŋ baɪ ɪksˈʧeɪnʤ] | Sort compares two consecutive (close) items in the list, and swap them if they are out of order. Each pass over the array results in the ascension of a bubble to its appropriate level of weight. Smallest element moves at the first place. Another name is Bubblesort | Сортировка обменом | |
|  | Bubblesort | [bʌblˈsɔːt] | Sort compares two consecutive (close) items in the list, and swap them if they are out of order. Each pass over the array results in the ascension of a bubble to its appropriate level of weight. Smallest element moves at the first place. Another name is Sorting by Exchange | Пузырьковая сортировка | |
|  | Shaker sort | [ˈʃeɪkə ˈsɔːt] | Is a bidirectional version of bubble sort.  Shaker sort unlike bubble sort orders the array in both directions. Every iteration of the algorithm consists of two phases. In the first one the lightest bubble ascends to the beginning of the array, in the second phase the heaviest bubble descends to the end of the array. Other names are cocktail sort, shake sort. | Шейкер-сортировка | |
|  | Cocktail sort | [ˈkɔkteɪl ˈsɔːt] | Is a bidirectional version of bubble sort.  Shaker sort unlike bubble sort orders the array in both directions. Every iteration of the algorithm consists of two phases. In the first one the lightest bubble ascends to the beginning of the array, in the second phase the heaviest bubble descends to the end of the array. Other names are shaker sort, shake sort. | Двунаправленная сортировка | |
|  | Shake sort | [ʃeɪk ˈsɔːt] | Is a bidirectional version of bubble sort.  Shaker sort unlike bubble sort orders the array in both directions. Every iteration of the algorithm consists of two phases. In the first one the lightest bubble ascends to the beginning of the array, in the second phase the heaviest bubble descends to the end of the array. Other names are cocktail sort, shaker sort. | Сортировка перемешиванием | |
|  | Partition Sort | [pɑːˈtɪʃn ˈsɔːt] | Sorts an array effectively by dividing the array into smaller and smaller parts, and sorting the smaller part in turn. Another name is Quicksort | Сортировка с разделением | |
|  | Quicksort | [ˈkwɪksɔːt] | Sorts an array effectively by dividing the array into smaller and smaller parts, and sorting the smaller part in turn. Another name is Partition Sort | Быстрая сортировка | |
|  | Median | [ˈmiːdjən] | The median of n items is defined as that item which is less or equal to half of the n items and which is larger or equal to the other half of the n items | Медиана | |
|  | Array | [əˈreɪ] | Array is a static, fundamental data structure, which is used to store of elements. Each element in the array gets its own space in the array. Any element can be accessed directly. | Массив | |
|  | Linked list | [lɪŋkt lɪst] | Is the set of data elements, each containing a link to its successor (and sometimes its predecessor). | Связанный список | |
|  | Singly Linked List | [ˈsɪŋgl lɪŋkt lɪst] | Is the data structure, where each node stores the contents of the node and a pointer or reference to the next node in the list. The last node has a pointer to nothingness to indicate that it is the last node. | Односвязный список | |
|  | Pointer | [ˈpɔɪntə] | Is the programming language object, whose value refers to (or "points to") another value stored elsewhere in the computer memory using its memory address. | Указатель | |
|  | Ordered linked list | [ˈɔːdəd lɪŋkt lɪst] | Is the ordered set of data elements | Упорядоченный линейный список | |
|  | Circular linked list | [ˈsɜːkjʊlə lɪŋkt lɪst] | Is the data structure, where all nodes are linked in a continuous circle, without using null. | Циклический список | |
|  | Doubly linked list | [ˈdʌblɪ lɪŋkt lɪst] | Is the data structure, where each node has a reference to both the next and previous nodes in the list. The last node links to nothing i.e., there are no nodes after it. Also, there are no nodes before the first node. | Двусвязный список | |
|  | Stack | [stæk] | Is the container of objects that are inserted and removed according to the last-in first-out (LIFO) principle. | Стек | |
|  | Backtracking | [ˈbæktrækɪŋ] | Is the process when you need to access the most recent data element in a series of elements. | Перебор с возвратом | |
|  | Palindrome | [ˈpælɪndrəʊm] | Is a word, phrase, number, or other sequence of characters which reads the same backward as forward, such as madam or racecar. | Палиндром | |
|  | Infix notation | [ˈɪnfɪks nəʊˈteɪʃn] | Is the notation commonly used in arithmetical and logical formulas and statements. It is characterized by the placement of operators between operands – "infixed operators" – such as the plus sign in "2 + 2". | Инфиксная нотация | |
|  | Reverse Polish notation (RPN) | [rɪˈvɜːs ˈpɔlɪʃ nəʊˈteɪʃn] | Is the mathematical notation in which every operator follows all of its operands. It is another name is postfix notation. | Обратная польская нотация | |
|  | Postfix notation | [pəʊstˈfɪks nəʊˈteɪʃn] | Is the mathematical notation in which every operator follows all of its operands. It is another name is Reverse Polish notation (RPN). | Постфиксная нотация | |
|  | Queue | [kjuː] | Is a container of objects (a linear collection) that are inserted and removed according to the first-in first-out (FIFO) principle | Очередь | |
|  | Bin sort | [bɪn sɔːt]] | Is the sorting algorithm that works by distributing the elements of an array into a number of buckets. Each bucket is then sorted individually, either using a different sorting algorithm, or by recursively applying the bucket sorting algorithm. Another name is Bucket sort | Бинарная сортировка | |
|  | Bucket sort | [ˈbʌkɪt sɔːt] | Is the sorting algorithm that works by distributing the elements of an array into a number of buckets. Each bucket is then sorted individually, either using a different sorting algorithm, or by recursively applying the bucket sorting algorithm. Another name is Bin sort | Блочная сортировка | |
|  | Priority queue | [praɪˈɔrɪtɪ kjuː] | Is the abstract data type which is like a regular queue or stack data structure, but where additionally each element has a "priority" associated with it. In a priority queue, an element with high priority is served before an element with low priority. If two elements have the same priority, they are served according to their order in the queue. | Очередь приоритетов | |
|  | Abstract data type (ADT) | [ˈæbstrækt ˈdeɪtə taɪp] | Is the mathematical model for data types where a data type is defined by its behavior (semantics) from the point of view of a user of the data, specifically in terms of possible values, possible operations on data of this type, and the behavior of these operations. | Абстрактный тип данных | |
|  | Double ended queue | [dʌbl ˈendɪd kjuː] | Is the linear list in which insertions and deletions are possible at either end | Двунаправленная очередь | |
|  | Binary tree | [ˈbaɪnərɪ triː] | Is a set of elements (nodes) which either is empty or consists of a root (node) with two binary trees called the left and the right subtree. | Двоичное дерево | |
|  | Tree | [triː] | Is the data structure implementing this ADT—that simulates a hierarchical tree structure, with a root value and subtrees of children with a parent node, represented as a set of linked nodes. | Дерево | |
|  | Root | [ruːt] | Is the top node in a tree. | Корень | |
|  | Child | [ʧaɪld] | Is a node directly connected to another node when moving away from the Root | Сын (ребенок) | |
|  | Parent | [ˈpɛərənt] | Is the converse notion of a child | Родитель | |
|  | Siblings | [ˈsɪblɪŋ] | Is a group of nodes with the same parent | Братья и сестры | |
|  | Descendant | [dɪˈsendənt] | Is a node reachable by repeated proceeding from parent to child | Потомок | |
|  | Ancestor | [ˈænsɪstə] | Is a node reachable by repeated proceeding from child to parent | Предок | |
|  | Leaf | [liːf] | Is a node with no children. Another name is External node | Лист | |
|  | External node | [eksˈtɜːnl nəʊd] | Is a node with no children. Another name is Leaf | Внешний узел | |
|  | Internal node | [ɪnˈtɜːnl nəʊd] | Is a node with at least one child | Внутренний узел | |
|  | Degree | [dɪˈgriː] | Is the number of sub trees of a node | Степень | |
|  | Edge | [eʤ] | Is the connection between one node and another | Ребро | |
|  | Path | [pɑːθ] | Is a sequence of nodes and edges connecting a node with a descendant | Путь | |
|  | Level | [levl] | Is the level of a node is defined by 1 + (the number of connections between the node and the root). | Уровень | |
|  | Height of node | [haɪt ɔv nəʊd] | Is the height of a node is the number of edges on the longest path between that node and a leaf | Высота узла | |
|  | Height of tree | [haɪt ɔv triː] | Is the height of a tree is the height of its root node | Высота дерева | |
|  | Depth | [depθ] | Is the depth of a node is the number of edges from the tree's root node to the node | Глубина | |
|  | Forest | [ˈfɔrɪst] | Is a forest is a set of n ≥ 0 disjoint trees | Лес | |
|  | Node | [nəʊd] | Is a structure which may contain a value or condition, or represent a separate data structure | Узел | |
|  | Subtree | [səbˈtriː] | Is the tree consisting of a node in T and all of its descendants in T | Поддерево | |
|  | Full binary tree | [fʊl ˈbaɪnərɪ triː] | Is a binary tree in which each node has exactly zero or two children | Полное двоичное дерево | |
|  | Complete binary tree | [kəmˈpliːt ˈbaɪnərɪ triː] | Is a binary tree, which is completely filled, with the possible exception of the bottom level, which is filled from left to right | Законченное двоичное дерево | |
|  | Perfectly balanced tree | [ˈpɜːfɪktlɪ ˈbælənst triː] | Is a tree, where for each node the numbers of nodes in its left and right subtrees differ by at most 1 | Идеально сбалансированное дерево | |
|  | Binary search tree | [ˈbaɪnərɪ sɜːʧ triː] | This tree has the following property: all children to the left of the node have values smaller than the value of a given node, and all children to the right of the node have values greater than the value of a given node. | Двоичное дерево поиска | |
|  | Traversal | [ˈtrævɜːsl] | Is the process that visits all the nodes in the tree | Обход | |
|  | Pre-order traversal | [priː-ˈɔːdə ˈtrævɜːsl] | Is the traversal in which each parent node is traversed before its children | Префиксный обход | |
|  | Post-order traversal | [pəʊst -ˈɔːdə ˈtrævɜːsl] | Is the traversal in which the children are traversed before their respective parents are traversed | Постфиксный обход | |
|  | In-order traversal | [ɪn -ˈɔːdə ˈtrævɜːsl] | Is the traversal in which a node's left subtree, then the node itself, and finally its right subtree are traversed | Прямой (инфиксный) обход | |
|  | Graph | [græf] | Is a structure consisting of a set of vertices \{v_1, v_2,\dots,v_n\} and a set of edges \{e_1, e_2,\dots,e_m\} | Граф | |
|  | Graph edge | [græf eʤ] | Is the pair of vertices \{v_i, v_j\}\ i,j \in \{1..n\} | Ребро графа | |
|  | Directed graph (or digraph) | [dɪˈrektɪd græf] | Is a graph that is a set of vertices connected by edges, where the edges have a direction associated with them | Направленный граф | |
|  | Undirected graph | [ʌndɪˈrektɪd græf] | Is a graph in which edges have no orientation. | Ненаправленный граф | |
|  | Oriented graph (orgraph) | [ˈɔːrɪentɪd græf] | Is a graph that is a set of vertices connected by edges, where the edges have a direction associated with them | Ориентированный граф | |
|  | Mixed graph | [mɪkst græf] | Is a graph in which some edges may be directed and some may be undirected | Смешанный граф | |
|  | Multigraph | [ˈmʌltɪgrɑːf] | Is a graph which have multiple edges (also called parallel edges), that is, edges that have the same end nodes | Мультиграф | |
|  | Path in a graph | [pɑːθ ɪn ə græf] | Is a sequence of vertices, such that there exists an edge or arc between consecutive vertices | Путь в графе | |
|  | Cycle | [saɪkl] | Is a path in which start and end vertices are the same | Цикл | |
|  | Weighted graph | [ˈweɪtɪd græf] | Is a graph in which every edge is associated with a real number, called edge weight | Взвешенный граф | |
|  | Degree of a vertex | [dɪˈgriː ɔv ə ˈvɜːteks] | Is the number of edges incident on it | Степень вершины | |
|  | Connected graph | [kəˈnektɪd græf] | Is an undirected graph in which every unordered pair of vertices in the graph is connected | Связанный граф | |
|  | Strongly connected graph | [ˈstrɒŋlɪ kəˈnektɪd græf] | Is a directed graph in which every ordered pair of vertices in the graph is strongly connected | Сильно связанный граф | |
|  | Bipartite graph | [baɪˈpɑːtaɪt græf] | Is a graph in which the vertex set can be partitioned into two sets, W and X, so that no two vertices in W share a common edge and no two vertices in X share a common edge | Двудольный граф | |
|  | Adjacency matrix | [əˈʤeɪsənsɪ ˈmeɪtrɪks] | Is a square matrix used to represent a finite graph. The elements of the matrix indicate whether pairs of vertices are adjacent or not in the graph | Матрица смежности | |
|  | Adjacency list | [əˈʤeɪsənsɪ lɪst] | Is a data structure stored a list of vertices, which are adjacent to current one. | Список смежности | |
|  | Unoriented incidence matrix | [ʌnˈɔːrɪentɪd ˈɪnsɪdəns ˈmeɪtrɪks] | Is a n × m matrix, where n and m are the numbers of vertices and edges respectively, such that b[i.j] = 1 if the vertex v\_i and edge x\_j are incident and 0 otherwise. | Матрица инцидентности неориентированного графа | |
|  | Incidence matrix of a directed graph | [ˈɪnsɪdəns ˈmeɪtrɪks ɔv ə dɪˈrektɪd græf] | Is a n × m matrix [b\_{ij}] where n and m are the number of vertices and edges respectively, such that b\_{ij} = -1 if the edge x\_j leaves vertex v\_i, 1 if it enters vertex v\_i and 0 otherwise. | Матрица инцидентности ориентированного графа | |
|  | Reachability | [rɪˈækəbɪlɪtɪ] | Reachability refers to the ability to get from one vertex to another within a graph. | Достижимость | |
|  | Reachability matrix | [rɪˈækəbɪlɪtɪ ˈmeɪtrɪks] | Matrix in which an entry of 1 in row i, col j indicates a path (with one or more edges) from I to j, and an entry of 0 means no path at all. | Матрица достижимости | |
|  | Eulerian graph | [ˈjuːlɪərɪən græf] | Is a graph containing an Eulerian cycle | Эйлеров граф | |
|  | Eulerian cycle | [ˈjuːlɪərɪən saɪkl] | Is a trail which starts and ends at the same graph vertex | Эйлеров цикл | |
|  | Euler path | [ˈɔɪlər pɑːθ] | Is a path that uses every edge of a graph exactly once. It starts and ends at different vertices | Эйлеров путь | |
|  | Acyclic graph | [əˈsaɪklɪk græf] | Is a graph with no cycles | Ациклический граф | |
|  | Spanning tree of a connected graph | [ˈspænɪŋ triː ɔv ə kəˈnektɪd græf] | Is a subgraph that contains all of that graph's vertices | Остовное дерево | |
|  | Minimum spanning tree (MST) of a graph | [ˈmɪnɪməm ˈspænɪŋ triː ɔv ə græf] | Is a spanning tree whose weight (the sum of the weights of its edges) is no larger than the weight of any other spanning tree | Остовное дерево минимальной стоимости | |
|  | Hamiltonian path | [ˈhæmɪltnɪən pɑːθ] | Is a path in an undirected or directed graph that visits each vertex exactly once | Гамильтонов путь | |
|  | Hamiltonian cycle | [ˈhæmɪltnɪən saɪkl] | Is a Hamiltonian path that is a cycle | Гамильтонов цикл | |