|  |  |  |
| --- | --- | --- |
| T(n) = aT(n/b) + f(n)  Not a valid formula because is not constant.  Kruskal’s algorithm and Prim’s algorithm. We can easily make each of them run in time  O (E lg V) using ordinary binary heaps. By using Fibonacci heaps, Prim’s algorithm runs in time O(E + V lg V), which improves over the binary-heap implementation if |V| is much smaller than |E|  The running time of Kruskal’s algorithm as O (E lg V).  A Turing machine is similar to a finite automaton with a supply of unlimited memory.  A configuration C1 yields a configuration C2 if the TM can legally go from C1 to C2 in a single computation step  Formally: suppose a, b, c ∈ , u, v ∈ \* and qi, qj ∈ Q.  1. We say that ua qi bv yields uac qj v if δ(qi, b) = (qj, c,R); (machine moves rightward)  2. We say that ua qi bv yields u qj acv if δ(qi, b) = (qj, c,L); (machine moves leftward)  A language is called Turing-decidable or simple decidable if some TM decides it.  Decide != Recognize  Any regular language is Turing-decidable.  Any context-free language is Turing-decidable.  Every decidable language is Turing-recognizable (a language is Turing-recognizable if it is recognized by a TM).  **For single-source shortest-paths problem**  Negative-weight edge and Negative-weight cycle reachable from s No  relax(u, v, w)  if v.d > u.d + w(u, v)  v.d = u.d + w(u, v)  v.Pie = u  A tree with n vertices has exactly n-1 edges  (|E| = |V| - 1)  **Complexity Bad to Good:** O(n!), O(2^n), O(n^2), O(n lg n), O(n), O(lg n), O(1).  Polynomial is always faster than (lg n).  **Adjacency Matrix, List:**  Matrix is better for performance  List is better for memory.  m[i,j] = { 0 if i = j  min {m[i,k] + m[k+1,j] +pi-1pkpj } if i<j  i<=k<j. | **True False Questions**  **F.** The set of states Q in a DFA can be empty.  **T**. A = { O' P > j } is not regular.  **T**. Let A, p, and s be defined as in the Pumping Lemma, if s = xyz, then s'= xz must also be in A.  **T**. Any language accepted by a DFA can also be accepted by a NFA.  **T**. The class of regular language is closed under the union operation.  **F**. A NFA is considered as a special case of a DFA.  **F.** If machine M has two input symbols that transit to q2 from qi, then M cannot be a DFA.  **T**. Every NFA has an equivalent DFA, and every DFA has an equivalent NFA.  **F**. Pumping Lemma is mostly used to prove that a language is actually regular.  **F.** If a DFA has n states, it cannot accept strings with a length greater than n2.  **F**. A NFA can have more than one state as the start state, a DFA can only have one state as the start state  **T**. If p = 1, vk> is a shortest path from vi to vk, 1 < i k, and A, = v El, v,> is a sub-path of p from v, to v1, then, pi, is a shortest path from v, to 121.  **T**. A relaxation step may decrease the value of the shortest-path cost d[v], and update v's predecessor it[v] if decrease step happens.  **F**. The single source shortest path of a graph forms a tree, and the tree is always unique.  **F**. In class we introduced at least two algorithms to find the single-source shortest-paths, and The Bellman-Ford one is fast.  **F**. Neither Bellman-Ford's nor Dijkstra's SSSP algorithm allows negative edges.  **F**. We cannot use Radix sort to sort strings.  **F.** Since Merge sort is the best comparison based algorithm and is O(n\*log n), Radix sort cannot be faster than that.  **3**. I have 10,000,000 positive integers stored in an array of integers (32-bit). To sort them using Radix sort, I need: (about 32\*(10,000,000 + 2) steps.)  **6.** I need to sort 1,800,000 numbers mite from -999,999 to 1,000,000, which of the following approach I should use the radix sort**:**  I can have an array of 2,000,000 bits, initialize to false, and offset a number with 999,999 as an index.  **F**. Quicksort is an excellent example for greedy algorithms and has a best time of 0(n log n).  **F**. By using Merge Sort, we can sort n2 elements in 0(lii to fin) time.  **T**. If f(n) = 0(n^6), then f(n) = 0(n^6 log n ).  **F**. The Master Theorem has three cases and it covers all possible recurrence situations.  **T**. If f(n) = 0(n6) and f(n) = Omega(n6) then f(n) = theta(n6).  **T**. An undirected, connected, and weighted graph may have more than one Minimum Spanning Tree (MST).  **F**. Adjacency Matrix cannot be used to represent a DAG.  **F**. Depth first search and breadth first search generally result in the same "Search tree."  **F**. Like Merge sort, topological sort-can sort n elements in 0(n log n) time.  **F.** Problems that can be solved by Dynamic Programming can, generally speaking, also be solved with a Greedy algorithm.  **T**. Kruskal and Prim algorithm in finding a MST are examples of Greedy algorithms.  **F**. If a connected graph G = (V, E) that satisfying the condition |V|=|E| - 1, then it is always a tree.  **T**. For a connected undirected graph G = (V, E), its Breadth-First Tree has IVI -I edges.  **T**. According to Parenthesis Theorem of DFS, two vertices do not have to be in the same tree.  **T**. Topologic sort can only be performed on DAGs, and the results on the same graph can be different.  **F**. The he 0-1 Knapsack Problem has a dynamic programming solution as well as a greedy algorithm solution.  **F.** The set of states Q in a DFA can be empty. When Q is not empty, the DFA can have more than one start state.  **F**. Let's A, p, and s be defined as in the Pumping Lemma, if s = xyz, then s'= xz must also be in A. In addition, the Pumping Lemma, for NFA is mostly used to prove that a language is actually regular T. Any language accepted by a NFA can also be accepted by a DFA.  **T**. The class of regular language is closed under the concatenation operation.  **F**. A NFA is considered as a special case of a DFA.  **F**. If machine M has two input symbols that transit to q2 from qi, then M cannot be a DFA.  **F**. Every CFG has an equivalent NFA, and every DFA has an equivalent CFG.  **F**. If a DFA has n states, it cannot accept strings with a length greater than n2.  **T**. A = {0"I" } can be accepted by a PDA.  HP-Hard does not have to be a Yes/No question. NP-Complete does.  ww is not context free. It is context sensitive  Pumping lemma is used to prove that a language is not regular or not context-free.  A Turing machine can do anything a computing device can do. | **Pumping lemma**: If A is a regular language, then there is a number *p* (the pumping length) where if *s* is any string in A of length at least *p*, then *s* may be divided into three pieces, s=xyz, satisfying the following conditions:  1.for each i≥0, ∈ A,  2.|y|>0, and  3.|xy|≤p.  **Union Example:**  **public void** printNLargestRecur(**int**[] a, **int** n){  **if**(!(n > a.**length** - 1)){  a = maxN(a, a.**length** - 1, a.**length** - (n + 1));  **for**(**int** i = a.**length** - 1; i > a.**length** - (n + 1); i--){  System.***out***.print(a[i] + **" "**);  }  } } **public int**[] maxN(**int**[] a, **int** s, **int** e){  //min section  /\***if**(s < e){  **for**(**int** i = s; i < a.**length**; i++){  **if**(a[s] > a[i])  swap(a, s, i);   minN(a, s + 1, e);  }}\*/  **if**(s > e)  {  **for**(**int** i = s; i > 0; i--){  **if**(a[s] < a[i])  swap(a, s, i);  minN(a, s - 1, e);  }} **return** a;}  If we use the U operator, then M accepts q1 or q2. If we use the concatenate operator, then M accepts q1 and q2.  An alphabet containing only 1’s is called a unary alphabet.  **Graph:**  If one vertex is undefined, then the whole graph is undefined.  SSSP is not necessarily unique.  Cannot have negative cycles that are reachable from “s”. But can have negative numbers.  Dijktras algorithm requires a DAG(Directed Acyclic Graph).  **Topological sort:**  Topological sort is not unique.  Topological sort doesn’t work on undirected graphs.  The node with the greatest finish time must finish first. Smallest finish time will finish last.   * *Graph G* = (*V*, *E*)   + *V* = set of vertices   + *E* = set of edges ⊆ (*V*×*V*) * Types of graphs   + Undirected: edge (*u*, *v*) = (*v*, *u*); for all *v*, (*v*, *v*) ∉ *E* (No self loops.)   + Directed: (*u*, *v*) is edge from *u* to *v*, denoted as *u* → *v*. Self loops are allowed.   + Weighted: each edge has an associated weight, given by a weight function *w* : *E* → **R**.   + Dense: |*E*| ≈ |*V*|2.   + Sparse: |*E*| << |*V*|2. * |*E*| = *O*(|*V|*2) |

E = {0^I 1^j | i>j}