Department of Computer Science and Engineering
National Sun Yat-sen University
Design and Analysis of Algorithms - Final Exam., Dec. 20, 2022

1. Explain each of the following terms. (28%)

   (a) NP, NP-complete
   (b) the node cover of a graph
   (c) set cover problem
   (d) skew heap
   (e) constrained 1-center problem
   (f) G^2 of a graph G
   (g) quadratic nonresidue problem

2. Please design an algorithm for finding both the minimum and the maximum of n elements with at most $\left\lfloor \frac{3n}{2} \right\rfloor$ comparisons. (12%)

3. Design a divide-and-conquer algorithm to solve the 2-D *maxima finding* problem. Then analyze the time complexity of the algorithm. (12%)

4. Explain the searching strategies: *depth-first search, breadth-first search* and *best-first search*. What data structures are used in these strategies? (12%)

5. The *first-fit* algorithm is a simple approximation algorithm for solving the *bin packing* problem. The algorithm puts an object into the ith bin as long as it is available and tries the (i+1)th bin if otherwise. Show that the number of bins used in the first-fit algorithm is no more than twice the number of bins needed in an optimal solution. (12%)

6. Prove that the *Hamiltonian decision* problem polynomially reduces to the *traveling salesperson decision* problem. (12%)

7. A sequence is a *palindrome* if it is the same by reading from either its forward or backward direction. For example **abbcbba** and **abccba** are palindromes. **abdbaa** is not a palindrome, but it can be decomposed into three palindromes **a**, **bdb**, **aa**, or two palindromes **abdba**, **a**. The latter is better than the former, since the number of decomposed palindromes is the minimum. Given a sequence $A=a_1a_2\ldots a_n$, the *palindrome decomposition* problem is to find the minimal number of decomposed palindromes for A. Let $d(i,j)$ denote the minimal number of decomposed palindromes for $a_ia_{i+1}\ldots a_j$, $1 \le i \le j \le n$. Furthermore, let $p(i,j)$ be a function for deciding whether $a_ia_{i+1}\ldots a_j$, $1 \le i \le n$, is a palindrome or not; $p(i,j)$ returns 1 if it is a palindrome, and returns 0 if it is not. Obviously, $d(i,j)$=1 and $p(i,j)$=1 if i=j. Based on $p(\ )$, please design the dynamic programming formula for determining $d(i,j)$ for any i, j. Note that you do not need to design $p(\ )$, you can use $p(\ )$ directly as a function call. (12%)

Answer:

1.

(a)

NP: The class of decision problem which can be solved by a non-deterministic polynomial algorithm.

NP-complete: The class of problems which are NP-hard and belong to NP.

(b)

In a graph G = (V, E), the node cover of a graph G is a set S, which is a subset of V, and S satisfies: for each edge (u, v) ∈ E, u ∈ S or v ∈ S.

(c)

$F = \{ S_i \} = \{ S_1, S_2, \dots, S_k \}$, $\bigcup_{Si \in F} Si = \{ u_1, u_2, \dots, u_n \}$.

T is a set cover of F if $T \subseteq F$ and $\bigcup_{Si \in T} Si = \bigcup_{Si \in F} Si$.

The set cover problem is to determine if T is a set cover of F.

(d)

A skew heap can be defined recursively as follows:

1. A heap with only one element is a skew heap (base case).
2. The result of merging two skew heaps is a skew heap.

(e)

Given a constrained line y=y’ and n points on 2D-plane, the constrained 1-center problem is to find a point whose y-coordinate is y’, and this point is the center of a circle which can enclose all points with minimum radius.

(f)

Given a graph G = (V, E), $G^2 = (V, E^2)$ is a graph that an edge $(u, v) \in E^2$ if there’s a path from u to v with at most 2 edges in G.

(g)

Given x and y’, for all y’ that GCD(x, y’) = 1, and $z^2 = y \bmod x$ for some z that $0 < z < x$ and GCD(x, z) = 1, y’ is a quadratic nonresidue if y’ $\neq$ y.

2.

Input : n個元素

Output: 元素中的最大值與最小值

Step 1: 將元素兩兩成對(Pairwise)互相比較，將較大的元素放入集合A，較小的放入集合B。

Step 2: 將A中的第一個元素令為Max，再依序與其他元素比較，若該元素較大時則取代為新的Max。

Step 3: 將B中的第一個元素令為min，再依序與其他元素比較，若該元素較小時則取代為新的min。

Max即所求的元素中最大值，min即所求的元素中最小值。

Step1 共比較$\frac{n}{2}$次

Step2 共比較$\frac{n}{2}-1$次

Step3 共比較$\frac{n}{2}-1$次

比較次數和為:$\frac{3}{2}n-2$次

3.

Input: A set S of n planar points.

Output: The maximal points of S.

Step 1: If S contains only one point, return it as the maximum. Otherwise, find a line L perpendicular to the x-axis which separates S into S_L and S_R, with equal sizes. =>O(n)

Step 2: Recursively find the maximal points of S_L and S_R. =>O(nlogn)
Step 3: Find the largest y-value of S_R, denotes as y_R. Discard each of the maximal points of S_L if its y-value is less than or equal to y_R. =>O(n)
Time complexity: O(nlogn)

4.
Depth-first search: DFS is a traversal approach in which the traverse begins at the root node and proceeds through the nodes as far as possible until we reach the node with no unvisited nearby nodes. DFS uses Stack data structure.

Breadth-first search: BFS is a traversal approach in which we first walk through all nodes on the same level before moving on to the next level. BFS uses Queue data structure.

Best-first search: The idea of Best-first search is to use an evaluation function to decide which adjacent is most promising and then explore. Best-first search uses priority queue (heap) data structure.

5.
$S(a_i)$: item i 的尺寸

OPT: 最佳解所用的 bin 數

m: first-fit algo.所用的 bin 數

$C(B_i)$: 使用 first-fit algo.在 bin B_i 內所裝的總量

OPT >= $\sum_{i=1}^{n} S(ai)$

$C(B_i) + C(B_{i+1}) > 1$

$C(B_1)+C(B_2)+\ldots+C(B_m) > m/2$
m < 2 * $\sum_{i=1}^{m} C(Bi)$ = 2 * $\sum_{i=1}^{n} S(ai)$ <= 2 * OPT
所以 m < 2 * OPT

6.
For a given graph G = (V, E), the Hamiltonian cycle problem is to find whether G contains a Hamiltonian cycle which passes through all the vertices of the graph exactly once.
For a given graph G' = (V, E'), with non-negative weights, the TSP is to find whether G' contains a simple cycle of length <= k that passes through all the vertices exactly once.

Set weight of edge (u,v): w(u,v)=0 if (u,v) $\in$E; w(u,v)=1 if (u,v) $\notin$E
The edges in G were weighted 0. In G', we set weight 1 to each edge that is not in graph G.
The method can be done in polynomial time.

The graph G has a Hamiltonian cycle iff there exists a cycle in G' passing through all vertices exactly once, and that has a length <= 0.
=>
If there exists a Hamiltonian cycle in the graph G, it forms a cycle in G' with length = 0, since weights of all the edges is 0. Hence there exists a solution for TSP in G' with length <= 0.

<=
If there is a cycle that passes through all vertices exactly once, and has length <= 0 in graph G', the cycle contains only edges that were originally presented in graph G. Hence there exists a Hamiltonian cycle in G.

7.
$$d(i,j) = \begin{cases} 1, & if\ \ p(i,j) = 1 \\ \min\limits_{i \le k \le j-1} \{d(i,k) + d(k+1,j)\}, & if\ \ p(i,j) = 0 \end{cases}$$