Department of Computer Science and Engineering
National Sun Yat-sen University
Design and Analysis of Algorithms - Final Exam., Jan. 7, 2020

1. Explain each of the following terms. (16%)
   (a) NP, NP-complete
   (b) G^2 of a graph G
   (c) L_∞ distance
   (d) the *pair-wise independent* property in the *move-to-front* heuristics for the self-organizing sequential search
2. Design a divide-and-conquer algorithm to solve the 2-D *maxima finding* problem. Then analyze the time complexity of the algorithm. (12%)
3. Present an algorithm for solving the *minimum spanning tree* problem on graphs. Analyze the time complexity of your algorithm. (12%)
4. Given n matrices A_1, A_2, …, A_n with size $p_0 \times p_1$, $p_1 \times p_2$, $p_2 \times p_3$, …, $p_{n-1} \times p_n$, respectively, design a *dynamic programming* method to determine the multiplication order of $A_1 \times A_2 \times \ldots \times A_n$ such that the number of scalar multiplications is minimized. Note that the computing of $A_i \times A_{i+1}$ requires $p_{i-1}p_ip_{i+1}$ scalar multiplications. (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. y is a *quadratic residue* mod x if there exists some z such that $z^2 \equiv y \bmod x$, where $0 < z < x$, GCD(x, z) = 1 and GCD(x, y) = 1. Suppose that you are given x = 9. Please decide whether each of the following y is a *quadratic residue* mod x or not. If it is, please present the value of z; otherwise, explain the reason. (12%)
   (a) y = 7.
   (b) y = 8.
7. Prove that the *partition decision* problem polynomially reduces to the 0/1 *knapsack decision* problem. (12%)
8. Suppose we are given n numbers a_1, a_2, …, a_n, each of which is a positive integer. Each a_i can be represented by its binary form with m bits as $a_i = a_{i,m-1}a_{i,m-2}\ldots a_{i1}a_{i0}$, where each a_{ij} is a binary bit, either 0 or 1. Now, we desire to subtract a value k

(may be 0) from one of the n numbers such that for each bit position j, $\sum_{i=1}^{n} a_{ij}$, $0 \le j \le m\text{-}1$, is odd. Note that if k is greater than all of the n numbers, you can view it as no solution.

(a) What is the value of k for the 3 numbers 5, 7, and 9? (2%)

(b) What is the value of k for the 4 numbers 4, 5, 6, and 7? (2%)

(c) Design a general method to determine the value of k. (8%)

Design and Analysis of Algorithms - Final Exam., Jan. 7, 2020 參考解答

1(a).

NP：可以用non-deterministic polynomial algorithm解決的decision problem。

NP-complete：NP與NP-hard的交集。

1(b).

$G^2$of a graph G：與 G 有相同點數，G 中任兩點之間，若有長度不超過 2 的路徑相連時(一個點透過中間點，連到另一個點，謂之路徑長度為 2)，則在G^2將此兩點以 edge 相連起來。

1(c).

假設兩點的座標值為(x1, y1) 與 (x2, y2)，L_∞ distance 之計算方式如下：

L_∞ distance：$\sqrt[\infty]{(x1-x2)^\infty+(y1-y2)^\infty}$ = max{|x1-x2|,|y1-y2|}.

1(d).

2.

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.
Step 2 :
Recursively find the maximal points of S_L and S_R.
Step 3 :
Find the largest y-value of S_R denoted as y_R. Discard each of the maximal points of S_L if its y-value is less than or equal to y_R.

Time complexity: T(n)
Step 1: O(n)
Step 2:

$$T(n)=\begin{cases} 2T\left(\frac{n}{2}\right)+O(n)+O(n) & ,n>1 \\ 1 & ,n=1 \end{cases}$$

$T(n) = 2T\left(\frac{n}{2}\right) + n + n$

$= 2(2\ T\left(\frac{n}{4}\right) + \frac{n}{2}+\frac{n}{2}) + 2n$

$= 4T(\frac{n}{4}) + 2n + 2n$

$= \ldots$

$= 2^k T\left(\frac{n}{2^k}\right) + 2nk$ (Assume $n = 2^k$,k = log n)

$= nT(1) + 2n \log n$

$= O(n \log n)$

Step 3: O(n)

故 2-D *maxima finding* problem 的 time complexity 為 O(n log n)。

3.

假設有一個無向的完全圖 G=(V,E)，令 A 集合為一空集合，開始將每個 vertex 納入 A 集合中，每次選取的 vertex 都要不在 A 集合裡但屬於 V，之後選取一條 edge(u,v)，u 在 A 集合裡，v 則不在 A 集合裡，且(u,v)的權重要最小，重複以上步驟直到所以 vertex 被納入 A 集合裡。假設|V|為 n，在選取完 edge 後，需要更新一次 u 保留的 edge，需要 O(n)的時間，recursive 的次數為(n-1)次，所以時間複雜度為 $O(n^2)$。

4.

假設 m(i,j)為計算 $A_i \times A_{i+1} \times \ldots \times A_j$ 的最小花費，那麼 m(i,j)根據 *dynamic programming* 可以表示成:m(i,j) = 0，if i = j 。

$m(i,j) = \min_{i \le k \le j-1} \{m(i,k)+m(k+1,j)+ p_{i-1}p_k p_j \}$，if i < j。

5.

$C(B_i)$:使用 first fit，Bin B_i裡面放的容量

$S(a_i)$: a_i所佔的容量

m : First fit bin 的數量

由 fist fit 演算法

$C(B_i) + C(B_i+1) > 1$

$C(1)+C(2)+\ldots.C(m) > \frac{m}{2}$

由 optimal (OPT) 演算法

$S(a_1)+S(a_2)+\ldots.+S(a_n)$ < OPT's Bin 的數量

又 C(1)+C(2)+….C(m) = S(a_1)+S(a_2)+….+S(a_n)

所以 $\frac{m}{2}$ < C(1)+C(2)+….C(m) = S(a_1)+S(a_2)+….+S(a_n) < OPT's Bin 的數量

$\frac{m}{2}$ < OPT's Bin 的數量

m < 2 OPT's Bin 的數量

First fit's bin 的數量 < 2 OPT's Bin 的數量

6.

$(\boldsymbol{a})$ $\boldsymbol{y = 7}$

$\boldsymbol{z^2 \equiv y \ mod \ x}$ ， $\boldsymbol{GCD(x,z)}$

列出所有可能的情形：

$1^2 \equiv 1 \ mod \ 9$ ， $GCD(9,1) = 1$

$2^2 \equiv 4 \ mod \ 9$ ， $GCD(9,2) = 1$

$4^2 \equiv 7 \ mod \ 9$ ， $GCD(9,4) = 1 \rightarrow true$

$5^2 \equiv 7 \ mod \ 9$ ， $GCD(9,5) = 1 \rightarrow true$

$7^2 \equiv 4 \ mod \ 9$ ， $GCD(9,7) = 1$

$8^2 \equiv 1 \ mod \ 9$ ， $GCD(9,8) = 1$

y=7 是 quadratic residue, 其中 z=4 或 z=5.

$(\boldsymbol{b})$ $\boldsymbol{y = 8}$

$\boldsymbol{z^2 \equiv y \ mod \ x}$ ， $\boldsymbol{GCD(x,z)}$

$1^2 \equiv 1 \ mod \ 9$ ， $GCD(9,1) = 1$

$2^2 \equiv 4 \ mod \ 9$ ， $GCD(9,2) = 1$

$4^2 \equiv 7 \ mod \ 9$ ， $GCD(9,4) = 1$

$5^2 \equiv 7 \ mod \ 9$ ， $GCD(9,5) = 1$

$7^2 \equiv 4 \ mod \ 9$ ， $GCD(9,7) = 1$

$8^2 \equiv 1 \ mod \ 9$ ， $GCD(9,8) = 1$

y=8 不是 quadratic residue, 因為 GCD(x,z), z=1~8, 沒有答案為 8

.

7.

instance of the partition problem：

S：$\{a_1, a_2, \dots, a_n\}$，

S 集合中是否存在一個Subset P

Such that $\sum_{a_i \in P} a_i = \sum_{a_i \notin P} a_i$

instance of 0/1 knapsack problem：

物品： n 項物品
重量 W：$\{w_1, w_2, \dots, w_n\}$
價值 P：$\{p_1, p_2, \dots, p_n\}$
背包重量限制 $\sum w_i x_i \leq M \quad x_i = 0 \ or \ 1, 1 \leq i \leq n$
目標：使得$\sum p_i x_i$得到最大值

partition problem $\Rightarrow$ 0/1 knapsack problem:
Let
$w_i = a_i \ , 1 \leq i \leq n$
$p_i = a_i \ , 1 \leq i \leq n$

$M = \frac{1}{2}\sum_{i=1}^{n} a_i$

$k = \frac{1}{2}\sum_{i=1}^{n} a_i$

如果partition problem有解，則存在一組Subset P，且此P集合的元素總合$\sum_{a_i \in P} a_i = \frac{1}{2}\sum_{i=1}^{n} a_i$ 。在此情況下，我們可以選擇P內的元素作為0/1 knapsack problem 的解答，此時必定滿足最大總價值$k = M = \frac{1}{2}\sum_{i=1}^{n} a_i$ ，因為最大總價值為k 的話，則背包內總重量也等於M。換言之，如果partition problem 有解，則 0/1 knapsack problem 亦有解。

partition problem $\Leftarrow$ 0/1 knapsack problem:
若0/1 knapsack problem有解，則所選取的內容必然滿足$\sum w_i x_i = M$而且$\sum p_i x_i = k$，將此解答中，x_i等於 1 的部分收集起來，作為P相對應之內容，即$P = \{a_i | x_i = 1\}$，此時$\sum_{a_i \in P} a_i = \frac{1}{2}\sum_{i=1}^{n} a_i$，換言之，partition problem 亦有解。

Thus the partition decision problem polynomially reduces to the 0/1 knapsack decision problem

8.

(a) k=4

	3	*2*	*1*	*0*
5	0	1	0	1
7	0	1	1	1
9	1	0	0	1
# of 1s	**1**	**2**	**1**	**3**

經過計算後發現只有，第 2 個 bit 之總和不滿足題目給定之條件。

因此取$k = 2^2 = 4$

計算新的矩陣如下 ：

	3	*2*	*1*	*0*
5-4	0	0	0	1
7	0	1	1	1
9	1	0	0	1
# of 1s	**1**	**1**	**1**	**3**

每一 column 的總和都為奇數，所以 k=4 即為所求。

(b) k=7

	2	*1*	*0*
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
# of 1s	**4**	**2**	**2**

經過計算後，發現全部的 bit 之總和皆不滿足題目給定之條件。

因此取$k = 2^2 + 2^1 + 2^0 = 7$

計算新的矩陣如下 ：

	2	*1*	*0*
4	1	0	0
5	1	0	1
6	1	1	0
7-7	0	0	0
# of 1s	**3**	**1**	**1**

每一行的總和都為奇數，所以 k=7 即為所求。

(c)

由上面的範例可以觀察到當 column 的總和為偶數時，只要讓這些 column 的總和減去 1，就可以符合題目所求，但需要考慮借位情形。

Input: n positive number$a_1, a_2, \dots, a_n$

Output: no solution or the value k

Step 1: 計算$S_j = \sum_{i=1}^{n} a_{ij}$,$0 \le j \le m$,m為位元的數量

Step 2: j=0

Step 3: 檢查S_j。若為奇數則將k_j設為**0**；若為偶數則將k_j設為**1**。其中，若$S_j \le 0$ 表示需向S_{j+1}借位，此時需更新S_{j+1}之值。

Step 4: j= j+1。若 $j \le m$，則回到 Step 2。

Step 5: 計算k值。

Step 6: 驗算：檢查最大的兩個數，分別減去k之後，是否每個bit之總和均為odd。若其中一個滿足，則輸出k，否則無解(若減去k之後，變為負數，亦不滿足)。