Midterm Exam. -- Algorithms (2005/11/8)

1. Let KANP(k, j, Y) represent the problem:

$$\text{maximize} \quad \sum_{k \le i \le j} p_i x_i$$

$$\text{subject to} \quad \sum_{k \le i \le j} w_i x_i \le Y$$

$$x_i = 0 \text{ or } 1, k \le j$$

Then the 0/1 knapsack problem is KNAP(1, n, M). Let $g_j(c)$ be the value of an optimal solution to KNAP(j+1, n, c). Clearly, $g_0(M)$ is the value of an optimal solution to KNAP(1, n, M) and $g_n(c)$=0 for all c.

(a). Please find the recurrence relation of $g_j(c)$ and $g_{j+1}(c)$. (10%)

(b). Given n=3, M=10, and P_i and W_i as follows:

i	W_i	P_i
1	10	40
2	3	20
3	5	30

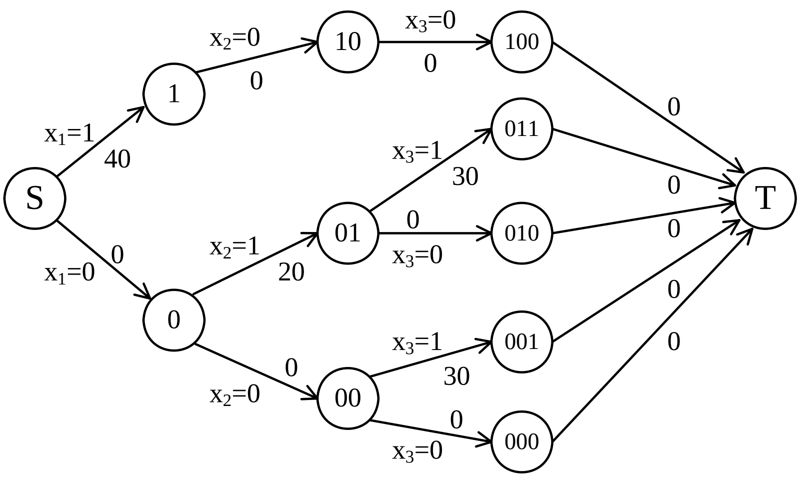
Please represent the above 0/1 knapsack problem KNAP(1, 3, 10) as a multistage graph. (10%)

Ans :

(a)

$$g_j(c)=\max\{g_{j+1}(c), g_{j+1}(c-w_{j+1})+p_{j+1}\}$$

(b)



2. A string is a sequence of symbols; for example, $X = <x_1,x_2,...,x_m>$ is a string of m symbols $x_1,x_2,...,x_m$. When we delete 0 or more symbols (not necessarily consecutive) from X, we get a subsequence of X. Write an algorithm using principle of optimality (dynamic programming) to calculate the length of the longest common subsequence of $X = <x_1,x_2,...,x_m>$ and $Y = <y_1,y_2,...,y_n>$. (15%) What is principle of optimality? (5%)

Ans :

(1)

➔**Let *L*[*i*, *j*] be the length of an LCS of the prefixes**
***Xi* = <*x*1, *x*2, … , *xi*> and *Yj* = <*y*1, *y*2, … , *yj*>,**
for 1 ≦ *i* ≦ *m* and 1 ≦ *j* ≦ *n*.

$$L[i, j] = \begin{cases} 0 & \text{if } i = 0, \text{ or } j = 0 \\ L[i-1, j-1] + 1 & \text{if } i, j > 0 \text{ and } x_i = y_j \\ \max(L[i, j-1], L[i-1, j]) & \text{if } i, j > 0 \text{ and } x_i \neq y_j \end{cases}$$

(2)

Principle of Optimality **: In an optimal sequence of decisions or choices, each subsequence must also be optimal.**

3. Write the Dijkstra's single-source shortest path algorithm (10%) and show that it is optimal in the worst case. (10%)

Ans :

(a)

```
Dijkstra (G, w, s) {                        //G=(V,E)
    for each vertex v ∈ V[G]    // Initialization
        do d[v] ← ∞                         //將所有的 s 到 v 的距離設為∞
           π[v] ← NIL
    d[s] ← 0                                //將 s 到 s 的距離設為 0
    S ← ∅                                   // S 為已確定最短路徑之 vertex 的集合
    Q ← V[G]
    while Q ≠ ∅
        do u ← Extract-Min(Q)                       //從 Q 中找出路徑為最短之
    vertex
            S ← S ∪ {u}                             //將 u 加入 S 中
            for each vertex v ∈ Adj[u]
            //對於每一個與 u 相鄰的 v，若 s 到 v 的距離比 s 經由 u 到 v 的
             距離長，則將最短路徑設為此
                if d[v] > d[u]+w(u,v)
                    then d[v] ← d[u]+w(u,v)
                         π[v] ← u                   //將 u 設為 v 的 predecessor
    }
```

(b) It can be easily seen that the worst-cast time complexity of Dijkstra's algorithm is $O(n^2)$ due to the repeated operations to calculate L(w). From another view of point, the minimum number of steps to solve the single-source shortest path problem is $\Omega(e)$ where e is the number of edges in the graph because every edge has to be examined. In the worst case, $\Omega(e)= \Omega(n^2)$. Therefore, in this sense, Dijkstra's algorithm is optimal.

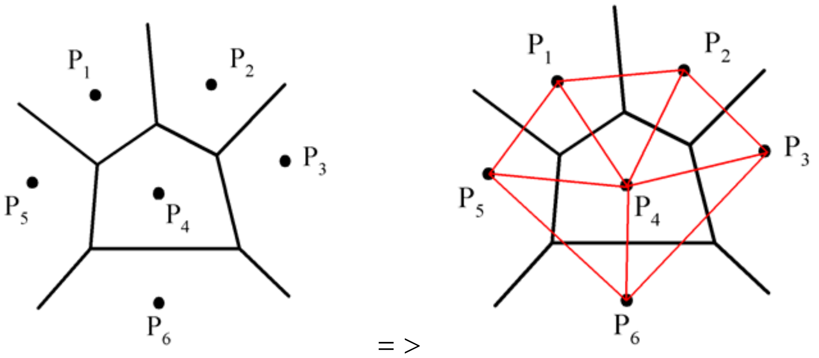
4. Let $T(n)$ be the time complexity of an algorithm and $T(n)=2T(n/2)+n$. Please represent $T(n)$ in the Big O notation. (10%)

Ans :

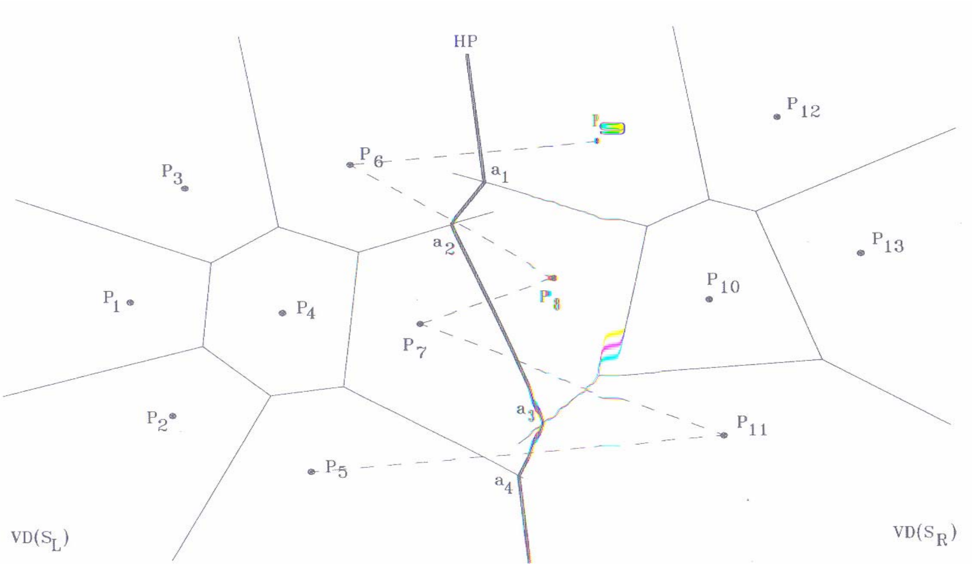
O(nlogn)

5. Given a Voronoi diagram for the 6 points below, please find the Delaunay

triangulation for the 6 points. (10%)



6. Please merge the following left and right Voronoi diagrams into one Voronoi diagram (10%)



7. A triomino is an L shaped object that can cover three squares of a chessboard. Please tile the following 8×8 defective chessboard with 21 triominos. (10%)