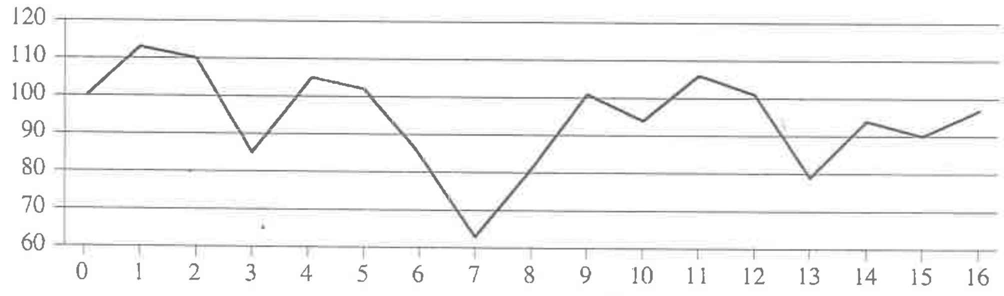
考試科目： Algorithms

1. Suppose that you been offered the opportunity to invest in the Volatile Chemical Corporation. Like the chemicals the company produces, the stock price of the Volatile Chemical Corporation is rather volatile. You are allowed to buy one unit of stock only one time and then sell it at a later date, buying and selling after the close of trading for the day. To compensate for this restriction, you are allowed to learn what the price of the stock will be in the future. Your goal is to maximize your profit. Figure 1 shows the price of the stock over a 17-day period. You may buy the stock at any one time, starting after day 0, when the price is $100 per share. Of course, you would want to "buy low, sell high"—buy at the lowest possible price and later on sell at the highest possible price—to maximize your profit. Unfortunately, you might not be able to buy at the lowest price and then sell at the highest price within a given period. In Figure 1, the lowest price occurs after day 7, which occurs after the highest price, after day 1.



Day	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Price	100	113	110	85	105	102	86	63	81	101	94	106	101	79	94	90	97
Change		13	−3	−25	20	−3	−16	−23	18	20	−7	12	−5	−22	15	−4	7

Figure 1: Information about the price of stock in the Volatile Chemical Corporation after the close of trading over a period of 17 days. The horizontal axis of the chart indicates the day, and the vertical axis shows the price. The bottom row of the table gives the change in price from the previous day.

(a) What is the major difference between "divide-and-conquer" and "dynamic programming"? (5%)

(b) Which technique should be applied to solve the problem? Please explain how to solve the problem in detail. (20%)

2. In the ***optimal binary search tree*** problem, we are given a sequence $K = <k_1, k_2, \ldots, k_n>$ of n distinct keys in sorted order (so that $k_1 < k_2 < \ldots < k_n$), and we wish to build a binary search tree from these keys. For each key k_i, we have a probability p_i that a search will be for k_i. Some searches may be for values not in K, and so we also have $n + 1$ "dummy keys" $d_0, d_1, d_2, \ldots, d_n$ representing values not in K. In particular, d_0 represents all values less than k_1, d_n represents all values greater than k_n, and for $i = 1, 2, \ldots, n - 1$, the dummy key d_i represents all values between k_i and k_{i+1}. For each dummy key d_i, we have a probability q_i that a search will correspond to d_i. Figure 2 shows a binary search trees for a set of $n = 5$ keys with the following probabilities:

i	0	1	2	3	4	5
p_i		0.15	0.10	0.05	0.10	0.20
q_i	0.05	0.10	0.05	0.05	0.05	0.10

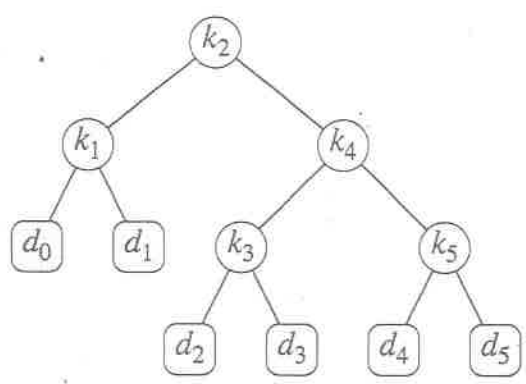


Figure 2: A binary search tree with expected search cost 2.80.

Each key k_i is an internal node, and each dummy key d_i is a leaf. Every search is either successful (finding some key k_i) or unsuccessful (finding some dummy key d_i). Because we have probabilities of searches for each key and each dummy key, we can determine the expected cost of a search in a given binary search tree T. Let us assume that the actual cost of a search equals the number of nodes examined, i.e., the depth of the node found by the search in T, plus 1. Then the expected cost of a search in T is

$$\begin{aligned} \text{E}[\text{search cost in } T] &= \sum_{i=1}^{n}(\text{depth}_T(k_i)+1)\cdot p_i + \sum_{i=0}^{n}(\text{depth}_T(d_i)+1)\cdot q_i \\ &= 1+\sum_{i=1}^{n}\text{depth}_T(k_i)\cdot p_i + \sum_{i=0}^{n}\text{depth}_T(d_i)\cdot q_i\,, \end{aligned}$$

where depth_T denotes a node's depth in the tree T. In Figure 2, we can calculate the expected search cost node by node:

node	depth	probability	contribution
k_1	1	0.15	0.30
k_2	0	0.10	0.10
k_3	2	0.05	0.15
k_4	1	0.10	0.20
k_5	2	0.20	0.60
d_0	2	0.05	0.15
d_1	2	0.10	0.30
d_2	3	0.05	0.20
d_3	3	0.05	0.20
d_4	3	0.05	0.20
d_5	3	0.10	0.40
Total			2.80

However, the tree in Figure 2 is not an optimal binary search tree. For the given set of probabilities, we wish to construct a binary search tree whose expected search cost is smallest. We call such a tree an optimal binary search tree.

(a) What is the major difference between "dynamic programming" and "greedy algorithm"? (5%)

(b) Which technique should be applied to solve the problem? Please explain how to solve the problem in detail. (20%)

3. Given a B tree with degree t,

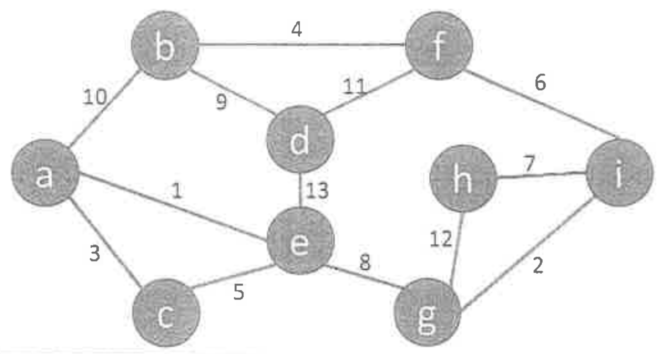
(a) What is the minimal number of children and keys for each internal node except the root node? (4%)

(b) Given an ordered data sequence: 30, 2, 10, 20, 80, 60, 70, 92, 100, 55, 77, please construct a B tree with $t = 2$ by inserting the data sequence. (6%)

(c) After (b), please delete the keys 30, 55, 80, 10 sequentially from the B tree and draw the final result. (10%)

4. Asymptotic notations are usually be used to quantify the time complexity in a systematic way. Please make a comparison of the following big-O notations: $O(n^2), O(\log n), O(2^n), O(n!), O(n\log n), O(n)$. (5%)

5. Given a connected weighted graph, please draw the minimal spanning trees by using Prim's and Kruskal's algorithms, respectively. (10%)



6. Given an ordered number sequence as follows: 50, 83, 10, 19, 5, 78, 39, 99, 77, 18, 1, 8.
   (a) Please construct a binary search tree by referring to the number sequence. (5%)
   (b) Based on the post-order sequence, please sort the numbers by insertion sort and merge sort, respectively. (You should write down all the steps) (10%)