【不可使用計算機】*作答前請先核對試題、答案卷(試卷)與准考證之所組別與考科是否相符！！

1. (6%) Given a binary search tree, T, as shown in Figure 1, split T at node 9 such that we have two trees *Small* and *Large*, and a node 9 (*Small* consists of nodes less than 9 and *Large* consists of nodes greater than 9). Draw the trees *Small* and *Large*.
2. (6%) Given two min-leftist trees shown in Figure 2, merge the two trees to get a new min-leftist tree. Draw the resulted tree.
3. (4%) Given the AVL tree shown in Figure 3, insert a node with key value 12.
4. (9%) Union/Find operation: Show that a tree obtained by applying the weighting rule for Union is bounded above by $O(\log n)$.

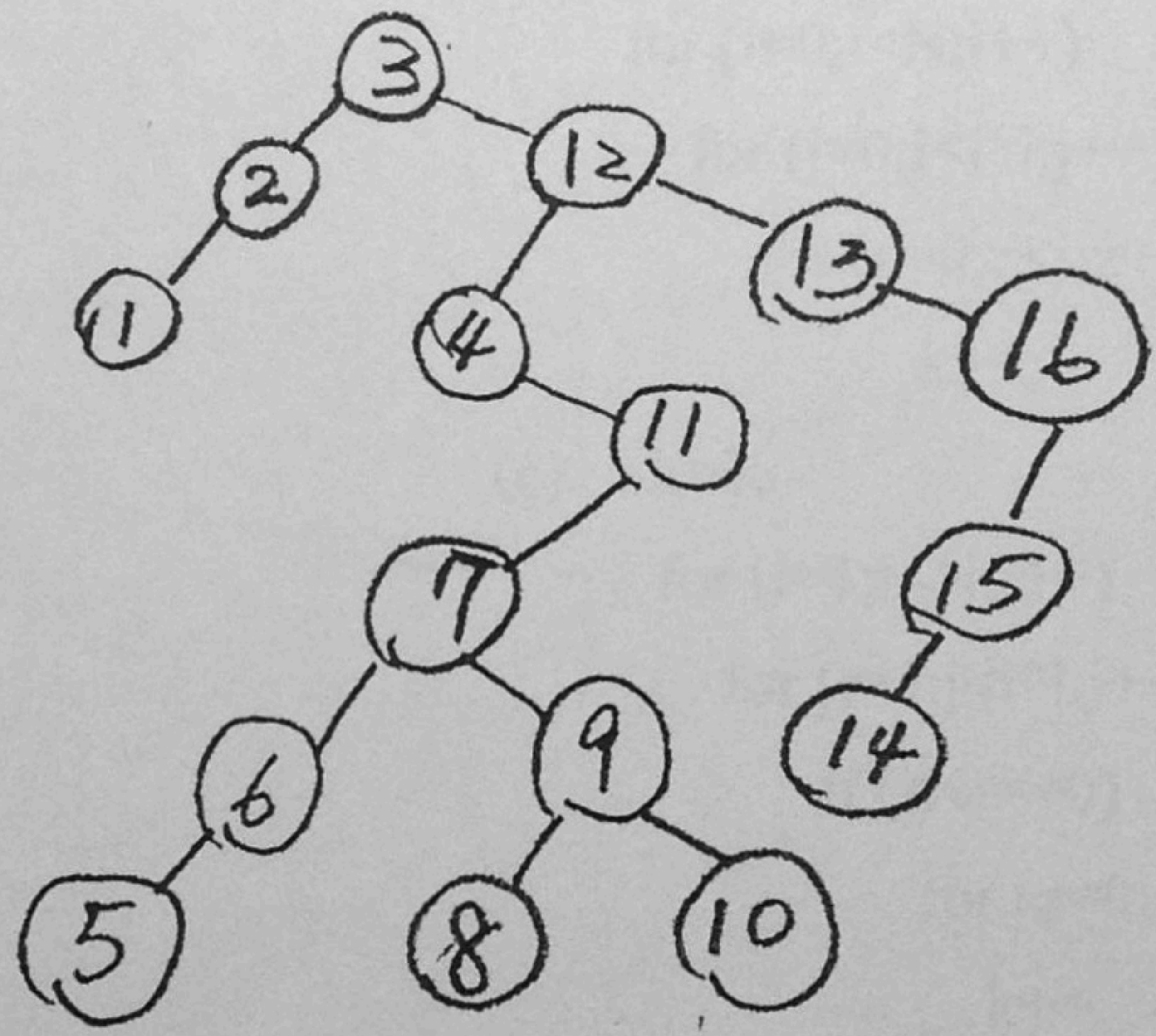


Figure 1

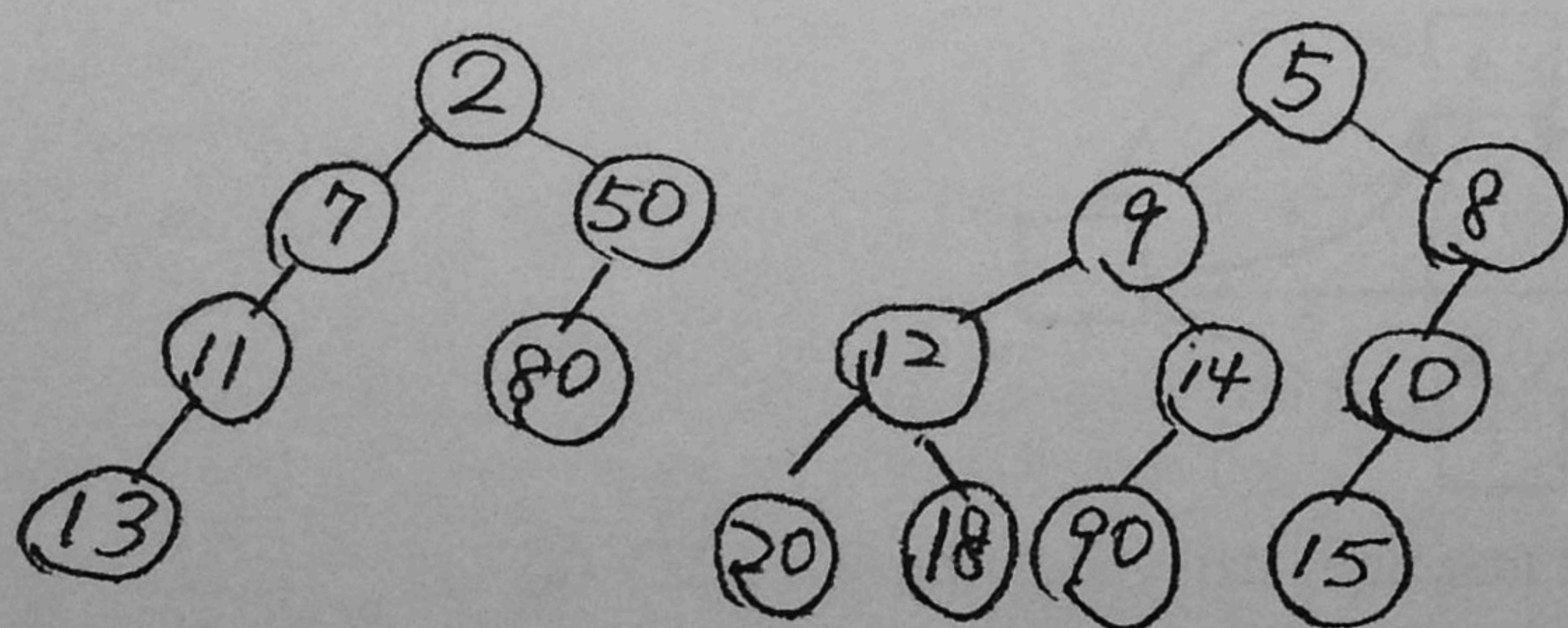


Figure 2

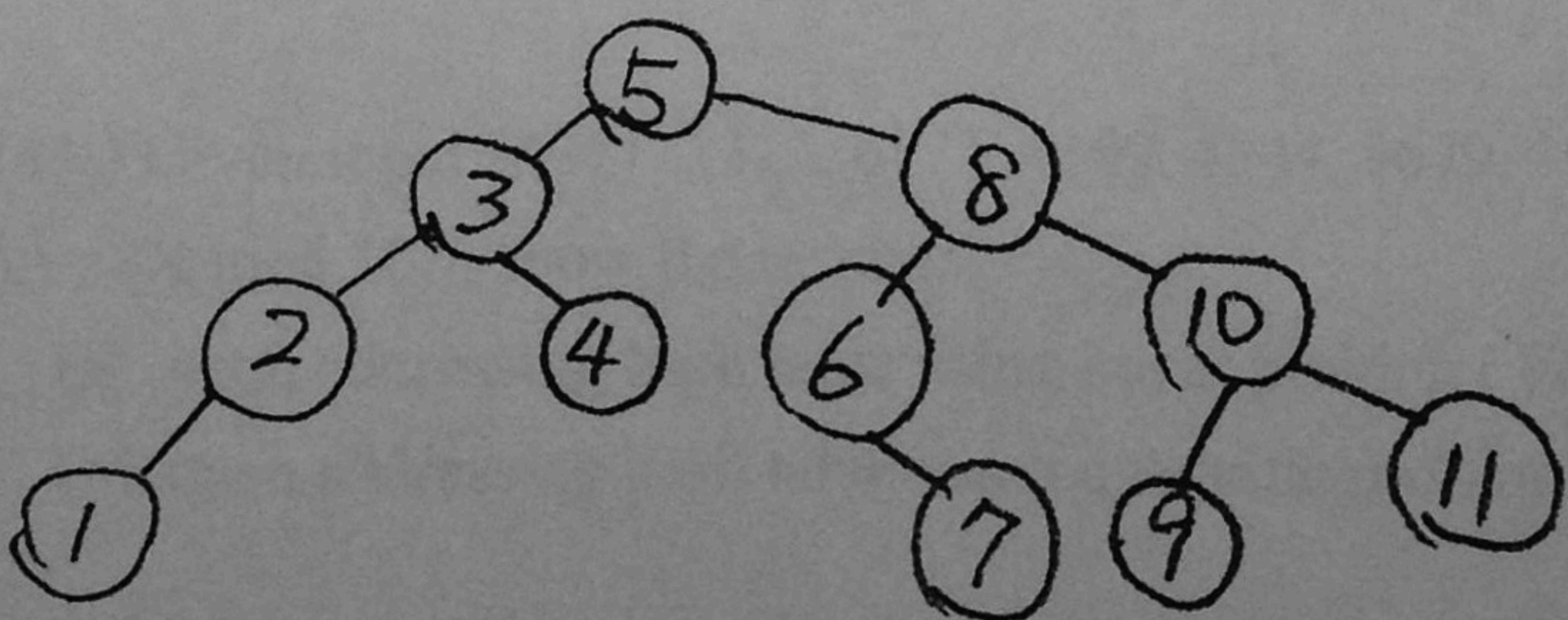


Figure 3

【不可使用計算機】＊作答前請先核對試題、答案卷(試卷)與准考證之所組別與考科是否相符！！

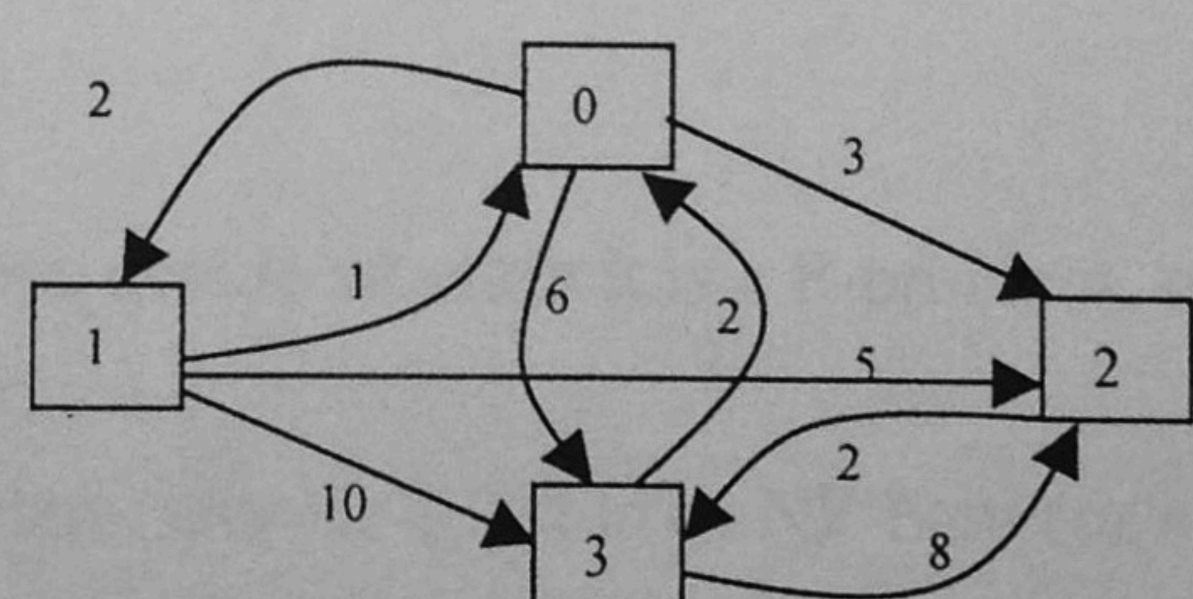
5. (9%) Please derive the corresponding time complexity (Big-Oh) for each of the following six program segments.

(a).
```
k=0;
for (i=0;i<N;i++)
    for (j=0;j<i;j++)
        k++;
```

(b).
```
k=0;
for (i=0;i<N;i++)
    for (j=0;j<i*i;j++)
        for (z=0;z<j;z++)
            k++;
```

(c).
```
k=0;
for (i=1;i<N;i++)
    for (j=1;j<i*i;j++)
        if (j%i==0)
            for (z=0;z<j;z++)
                k++
```

6. (12%) Given the following graph



(a) (3%) Use adjacency lists represent this graph.

(b) (4%) According to the adjacency lists obtained, output DFS(0) and BFS(0).

(c) (5%) Please derive all-pairs shortest-paths of the graph.

7. (4%) Given input {4371, 1323, 6173, 4199, 4344, 9679, 1989} and a hash function h(X) = (X mod 10)), show the results:

(a) open addressing hash table using linear probing (F(i) = i)

(b) open addressing hash table using quadratic probing ($F(i) = 3*i^2$)

8. (10%) Consider the following problems and algorithms.

(1) Merge sort for sorting problem

(2) Heap sort for sorting problem.

(3) Minimum spanning tree problem.

(4) Floyd-Warshall algorithm for all pairs shortest path problem.

(5) Finding a maximum independent set in an interval graph.

(6) Huffman's algorithm for Huffman code problem.

(7) Strassen's matrix multiplication algorithm.

(8) Constructing an optimal binary search tree for weighted searching problem, i.e to minimize the expected searching time.

(9) Longest Common subsequence problem.

(10) The Ford- Fulkerson method for maximum flow problem.

Each of the above problems (or algorithms) can be solved by (or classified as) one of the following design techniques

(a) Divide and Conquer.

(b) Dynamic Programming

(c) Greedy method

(d) Others (none of the above three techniques).

Choose the proper technique to each of the above problems (or algorithms)

Please just answer (a), (b), (c), or (d) for each of the above 10 problems.

9. (10%) Assume $P \neq NP$.

For each of the following problems, decide whether it is a P-problem, or an NP-hard (or NP-complete) problem, or neither..

Please answer 1, if it is a P-problem, answer 2 if it is an NP-hard (or NP-complete) problem, and answer 3 if it is neither 1 nor 2.

(1) Find a longest simple path between two nodes, where the given graph has positive edge weights.

(2) Find a shortest simple path between two nodes in a directed graph with negative and/or positive edge weights, and containing negative weight cycles.

(3) Find a negative weight directed cycle in a weighted directed graph.

(4) Find a positive weight directed cycle in a weighted directed graph.

(5) Find a largest cycle in a graph, where the edge-weight is 1 for each edge.

(6) Find a smallest cycle in a graph, where the edge-weight is 1 for each edge.

(7) Find a maximum cut in a flow network.

(8) Find a minimum cut in a flow network.

(9) Find a maximum independent set in an interval graph.

(10)The 2-CNF-Satisfiability problem.

【不可使用計算機】＊作答前請先核對試題、答案卷(試卷)與准考證之所組別與考科是否相符！！

10. (5%) Consider a two-position switch with two inputs and two outputs, as shown in Fig. A. In one position inputs 1 and 2 are connected to outputs 1 and 2, respectively. In the other position inputs 1 and 2 are connected to outputs 2 and 1, respectively. Using these switches and applying the divide and conquer technique, we can design a network with n inputs and n outputs which is capable of achieving any of the n! possible permutations of the inputs, as shown in Fig. B.

(1)Find the number of switches used in the networks for n=8. 2%

(2) For general $n=2^k$, find the number of switches used in the networks. (Write your answer in terms of n and/or k. Give the exact number, NOT just the asymptotic big O order.) 3%

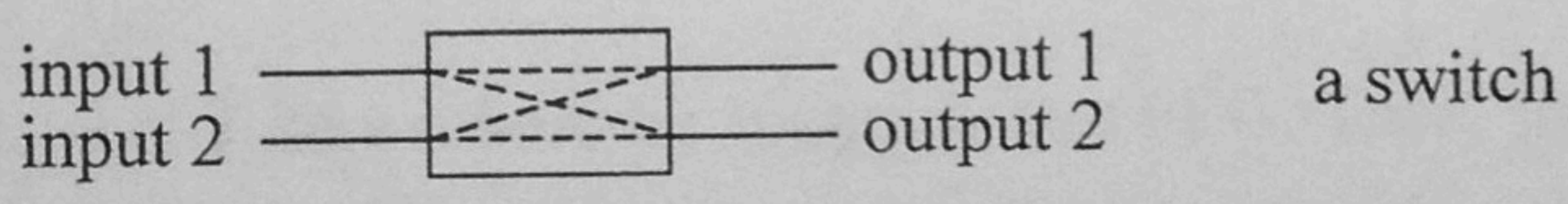


Fig.A

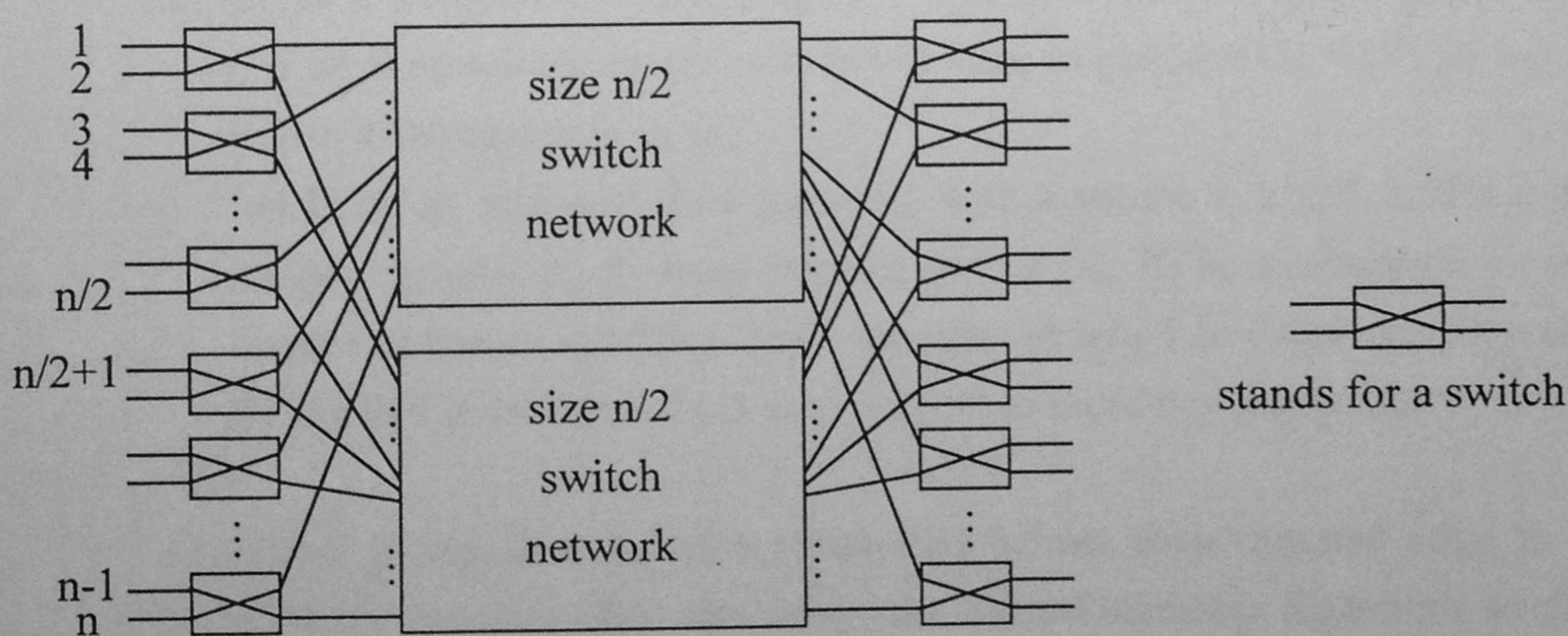


Fig.B

【不可使用計算機】＊作答前請先核對試題、答案卷(試卷)與准考證之所組別與考科是否相符！！

11. (15%) For each of the following statements to indicate whether the statement is true or false, respectively. If the statement is true, briefly state why. If the statement is false, explain why or give a counterexample. You will be deducted one point with wrong justification or without justification.

a. Let G = (V, E) be a directed graph with negative-weight edges, but no negative-weight cycles. Then, one can compute all shortest paths from a source $s \in V$ to all $v \in V$ faster than Bellman-Ford using the technique of reweighting.

b. Let G = (V, E) be a connected, undirected graph with edge weights, and assume all edge weights are distinct. Consider a cycle $\leq v_1, v_2, \ldots, v_k, v_{k+1} \geq$ in G, where $v_{k+1} = v_1$, and let (v_i, v_{i+1}) be the edge in the cycle with the largest edge weight. The (v_i, v_{i+1}) does not belong to the minimum spanning tree T of G.

c. Let T be a minimum spanning tree of G. Then, for any pair of vertices *s* and *t*, the shortest path from *s* to *t* in G is the path from *s* to *t* in T.

d. Given a weighted directed graph G = (V; E; *w*) and a shortest path *p* from *s* to *t*, if we doubled the weigh *w* to every edge to produce $G_0 = (V; E; w_0)$, then *p* is also a shortest path in G_0.

e. Let G be an arbitrary flow network, with a source *s*, a sink *t*, and a positive integer capacity c_e on every edge e; and let (A, B) be a minimum *s-t* cut with respect to these capacities. Now suppose we add 1 to every capacity; then (A, B) is still a minimum *s-t* cut with respect to these new capacities.

12. (10%) In the flow network illustrated below, each directed edge is labeled with its capacity. We are using the Ford-Fulkerson algorithm to find the maximum flow.

a. (5 points) Show every augmentation path (but you do not need to show the whole network to save time) using Edmonds–Karp algorithm. Draw the residual network after you have updated the flow using these augmenting paths.

b. (5 points) What is maximum flow? Draw a dotted line through the original graph to represent the minimum cut.