請使用答案卡作答

For each problemset, if your answer is correct for all the questions in the problemset, you receive the full points; or otherwise your answer is incorrect for any of the questions in the problemset, you receive 0 point. If we place a mark †† next to a problemset number, that means each question in the problemset may have one or more answers. If a question has multiple answers, then a correct response needs to contain all of them.

1. (5%) What are the outputs of the following codes? ___①___

```
#include <iostream>
#include <queue>
#include <stack>
using namespace std;
int main() {
    queue<int> que; stack<int> stk;
    que.push(13); que.push(2); que.push(5); que.push(27);
    stk.push(2); stk.push(32); stk.push(17); stk.push(63);
    cout << que.front() << ", " << que.back() << ", " << stk.top() << endl;
    return 0;
}
```

(A) 13, 27, 63 (B) 13, 27, 2 (C) 27, 13, 2 (D) 27, 13, 63 (E) 13, 2, 63

2. (5%) Let *head*($L$) be the first node of a linked list $L$, NIL is a null pointer, $x$ be a node in $L$, and *prev*[$y$] and *next*[$y$] be the pointers to the previous and the next nodes of $y$ on $L$, respectively. What is the function of the following pseudo codes? ___②___

```
if prev[x] ≠ NIL
    then next[prev[x]] ← next[x]
else
    head[L] ← next[x]
if next[x] ≠ NIL
    then prev[next[x]] ← prev[x]
```

(A) Inserting a node right in front of $x$. (B) Inserting a node right in back of $x$. (C) Deleting $x$. (D) Swapping $x$ and the node right in front of $x$. (E) Swapping $x$ and the node right in back of $x$.

3. †† (5%) What descriptions to the heap are true? You will get the score only if your answer is completely correct. ___③___
(A) The time complexity of heap sort is $O(n \log n)$. (B) Heap sort is faster than quick sort in the worst case. (C) If a binary heap is implemented by using an array, node $i$ is at the $i^{th}$ position of the array. In addition, the right child of node $i$ is at the $(i \times 2)^{th}$ position. (D) In a maximum heap, a parent node is smaller than its children. (E) If there are $n$ nodes in a heap, the height of the heap is $O(\log n)$.

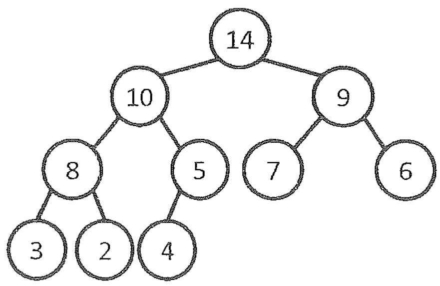
【不可使用計算機】＊作答前請先核對試題、答案卷(試卷)與准考證之所組別與考科是否相符！！

4. (5%) Without considering compiler optimization strategies, if the first and the second outputs of the following codes are 0x7ffd9e21bc00 and 0x7ffd9e21bc04, respectively, what will be the third output? __(4)__

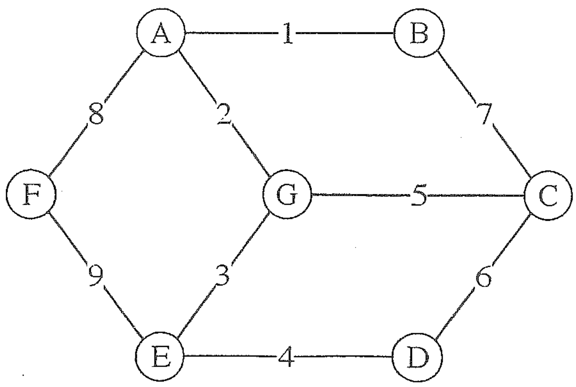
```
#include <iostream>
using namespace std;
int main() {
    int x[5][5];
    cout << &x[0][0] << " " << &x[0][1] << " " << &x[3][2] << endl;
    return 0;
}
```

(A) 0x7ffd9e21bc48 (B) 0x7ffd9e21bc44 (C) 0x7ffd9e21bc64 (D) 0x7ffd9e21bc30
(E) 0x7ffd9e21bc68

5. (5%) Suppose that the binary max heap below is implemented by using an array. When the root is removed from the heap, the system will update the heap structure immediately to maintain its property. After that, what will be the $4^{th}$ number in the array? __(5)__ Note that the root is the first number.
(A) 9 (B) 8 (C) 5 (D) 3 (E) 4




6. (5%) Consider the following graph. Please answer the following questions.



- ⑥ : What is the cost of the minimum cost spanning tree of the graph?
  (A) 27 (B) 21 (C) 25 (D) 20 (E) None of the above
- ⑦ : When Kruskal's algorithm is used, what is the weight of the last edge to be added into the minimum cost spanning tree of the above graph?
  (A) 6 (B) 7 (C) 8 (D) 9 (E) None of the above
- ⑧ : When Prim's algorithm is used and starts from node F, what is the weight of the last edge to be added into the minimum cost spanning tree of the above graph?
  (A) 5 (B) 6 (C) 7 (D) 8 (E) None of the above

7. (6%) Consider inserting the following numbers into an empty AVL tree. Please answer the following questions.
   20 30 40 50 60 70 10 5 25 45

   - ⑨ : What is the number in the leftmost node of the resultant AVL tree?
     (A) 5 (B) 10 (C) 20 (D) 25 (E) None of the above
   - ⑩ : What is the number in the root node of the resultant AVL tree?
     (A) 20 (B) 25 (C) 40 (D) 30 (E) None of the above
   - ⑪ : There are four types of rotations in AVL tree. What are the last two rotations performed during insertion of these numbers?
     (A) RR, RR (B) LL, LL (C) RL, LR (D) LR, RL (E) None of the above

8. (6%) Consider inserting the following numbers into an empty B-tree of order 3. Please answer the following questions.
   20 30 40 50 60 70 10 5 25 45

   - ⑫ : What is the sum of the numbers in the rightmost node of the resultant B-tree?
     (A) 120 (B) 130 (C) 110 (D) 115 (E) None of the above
   - ⑬ : What is the sum of the numbers in the root node of the resultant B-tree?
     (A) 25 (B) 40 (C) 30 (D) 55 (E) None of the above
   - ⑭ : How many node split operations are performed during insertion of these numbers??
     (A) 3 (B) 2 (C) 5 (D) 4 (E) None of the above

9. (10%) Consider the following C program. Please answer the following questions.

   - ⑮ : What is the value of the smallest element in array d? (A) 2 (B) 4 (C) 6 (D) 8 (E) None of the above
   - ⑯ : What is the value of the largest element in array d? (A) 6 (B) 12 (C) 18 (D) 24 (E) None of the above
   - ⑰ : What is the time complexity of the above program? (A) $O(\log N)$ (B) $O(N)$ (C) $O(N \log N)$ (D) $O(N^2)$ (E) None of the above

```
#include <stdbool.h>
#define N 8
#define START 4
int l[N][N]={{   0,1000,1000,1000,1000,1000,1000,1000},
             {   8,   0,1000,1000,1000,1000,1000,1000},
             {  10,   2,   0,1000,1000,1000,1000,1000},
             {1000,1000,   4,   0,1000,1000,1000,1000},
             {1000,1000,1000,  12,   0,   2,1000,1000},
             {1000,1000,1000,  10,1000,   0,   6,   8},
             {1000,1000,1000,1000,1000,1000,   0,  10},
             {   4,1000,1000,1000,1000,1000,1000,1000}};
bool s[N];
int d[N];
int main(void) {
  int v=START;
  for(int i=0;i<N;i++) {
    s[i]=false;
    d[i]=l[v][i];
  }
  s[v]=true;
  d[v]=0;
  for(int i=0;i<N-2;i++) {
    int u_max=2000;
    int u;
    for(int j=0;j<N;j++) {
      if (s[j]==false && d[j]<u_max) {
        u=j;
        u_max=d[j];
      }
    }
    s[u]=true;
    for(int w=0;w<N;w++)
      if (!s[w] && d[u]+l[u][w]<d[w])
        d[w]=d[u]+l[u][w];
  }
}
```

# 國立交通大學108學年度碩士班考試入學招生試題

科目：資料結構與演算法(1101)　　考試日期：108年2月13日 第1節

系所班別：資訊聯招　　

【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. (6%) Let $A$ be an array of integers. Consider the following algorithms.

**Algorithm 1:** P$(A, p, r)$

```
index = Random(p, r);
exchange A[index] with A[r];
x = A[r];
i = p - 1;
for j = p to r - 1 do
    if A[j] ≤ x then
        i = i + 1;
        exchange A[i] with A[j];
    end
end
exchange(A[i + 1], A[r]);
return i + 1;
```

**Algorithm 2:** Q$(A, p, r)$

```
1 if p < r then
2     repeat
3         q = P(A, p, r);
4     until the larger part is at most 2/3 of the subarray A[p..r];
5     Q(A, p, q - 1);
6     Q(A, q + 1, r);
7 end
```

- Algorithm Q has running time ____18____ in the average case (as close as possible).
  (A) $O(\log n)$ (B) $O(n)$ (C) $O(n \log n)$ (D) $O(n^2)$ (E) $O(\log^2 n)$
- ____19____ What is the probability that the repeat loop (lines 2-4) of Algorithm Q is executed exactly once?
  (A) 1/2 (B) 1/4 (C) 1/3 (D) 2/3 (E) 3/4
- ____20____ What is the average number of iterations that the repeat loop (lines 2-4) of Algorithm Q is executed?
  (A) 2 (B) 3 (C) 4 (D) 1.5 (E) 2.5

11. †† (6%) Given a sequence of $n$ positive numbers, we want to put $n-1$ pairs of parentheses around the $n$ numbers, such that the total sum of the $n-1$ intermediate sums is minimized.

For example, given fours positive numbers in order 4, 1, 2, 3, we can put three pairs of parentheses around and add them $((4+1)+(2+3)) = ((5)+(5)) = (10)$. Note that three intermediate sums are generated, namely 5, 5, and 10. The total sum of these three numbers is 5+ 5 + 10=20. If we put the parentheses differently as $(4+((1+2)+3))$, the three intermediate sums generated are 3, 6, and 10 and the sum is 19.

Consider the input: 4, 4, 8, 5, 4, 3, 5. Let the smallest intermediate sum be $X$.

- $X$ mod 10 is ____21____.
  (A) 1 (B) 2 (C) 3 (D) 4 (E) 5

- The integer part of $X/10$ is ___(22)___.
  (A) 6 (B) 7 (C) 8 (D) 9 (E) 5
- ___(23)___ Which statement about this question is correct?
  (A) It needs exponential time to find the answer.
  (B) This problem does not have an optimal substructure property.
  (C) The best solution is smaller than 50.
  (D) The best solution is smaller than 100.
  (E) It does NOT have an $O(n^2)$-time algorithm.

12. †† (6%) Let $G = (V, E)$ be a connected flow network with source $s$, sink $t$,and an integer capacity $c(e)$ on each edge $e \in E$. Let $C$ be the maximum capacity of the edges. Consider the following proposed algorithm.

**Algorithm 3:** Proposed_Max_Flow($G, s, t$)

```
1  C = max_{e∈E} c(e);
2  initialize flow f = 0;
3  K = 2^{⌊lg C⌋};
4  while K ≥ 1 do
5      while there exists an augmenting path p of capacity at least K do
6          augment f along p;
7      end
8      K=K/2 ;
9  end
10 return f;
```

- ___(24)___ Which of the following statements are (or is) true?
  (A) The capacity of a minimum cut is unique.
  (B) The minimum cut is not unique.
  (C) The minimum cut has capacity at most $C|E|$
  (D) It takes $O(V)$ time to find an augmenting path of capacity at least $K$, if one exists.
  (E) The inner while loop lines 5-6 is executed $VE$ times for each $K$.

- The running time of the proposed algorithm is ___(25)___.
  (A) $O(VE^2)$ (B) $O(V^2E)$ (C) $O(VE \lg C)$ (D) $O(E^2 \lg C)$ (E) $O(E \lg C)$

- ___(26)___ Which of the following statements are (or is) true?
  (A) If all the edge capacities are different, then there is a unique minimum cut.
  (B) If all the edge capacities are different, then there is a unique set of edge flows that gives the maximum flow value.
  (C) When we cannot find an augmenting path, let $S$ be the set of $s$ and the nodes reachable from $s$ in the residual flow network. Then $S$ and $V \backslash S$ form a minimum cut.
  (D) Let $(S, T)$ be a minimum cut corresponding to a maximum flow of $G$. Then all

the edges from $S$ to $T$ have zero residual capacity.
(E) With integral capacity on each edge, a maximum flow may have non-integral flow on some edge(s).

13. †† (8%) Given an undirected graph $G = (V, E)$ with $n = |V|, m = |E|$ and weights $w : E \to \mathbb{R}^+$, we define an order $v_1, \ldots, v_n$ of $V$ an Magic Order if for all $i \in \{2, \ldots, n\}$:

$$\sum_{e \in E(\{v_1,\ldots,v_{i-1}\},\{v_i\})} w(e) = \max_{j \in \{i,\ldots,n\}} \sum_{e \in E(\{v_1,\ldots,v_{i-1}\},\{v_j\})} w(e),$$

where $E(A, B)$ indicates the edges between disjoint vertex subsets $A$ and $B$. Consider the following incomplete algorithm.

**Algorithm 4:** : Magic_Order($G$)

```
1 set key(v) = 0 for all v ∈ V;
2 for i = 1 to n do
3 |   choose v_i from V\{v_1, ..., v_{i-1}} such that it has maximum key value
  |     (breaking ties arbitrarily);
4 |   for v ∈ V\{v_1, ..., v_i} do
5 |   |   key(v) = key(v)+ ______
6 |   end
7 end
```

- __(27)__ What is the missing part in line 5 in the above Algorithm?
(A) $\sum_{e \in E(\{v_1,\ldots,v_{i-1}\},\{v\})} w(e)$
(B) $\sum_{e \in E(\{v_1,\ldots,v_{i-1}\},\{v_j\})} w(e)$
(C) $\sum_{e \in E(\{v_{i-1}\},\{v\})} w(e)$
(D) $\sum_{e \in E(\{v_{i-1}\},\{v_i\})} w(e)$
(E) $\sum_{e \in E(\{v_i\},\{v\})} w(e)$

- __(28)__ Which of the following statements are (or is) true?
(A) It takes $O(n)$ time for line 3, if it is implemented with an array.
(B) If use a binary heap, then the cost of line 3 is $O(\sqrt{\log n})$.
(C) If use a Fibonacci heap, then the amortized cost of line 3 is $O(\log n)$.
(D) If use a Fibonacci heap, then line 5 has the amortized cost $O(1)$.
(E) If use a binary heap, then line 5 has worst case time complexity $O(\log n)$.

- With Fibonacci Heap, the above algorithm has amortized cost __(29)__. (pick one as close as possible)
(A) $O(n \log n)$ (B) $O(m \log n)$ (C) $O(n + m \log n)$ (D) $O(m + n \log n)$ (E) $O(m \log m)$

14. †† (6%) Let $G$ be a connected simple undirected graph of at least 6 nodes, and let $T$ be a BFS tree of $G$ rooted at node $r$. Answer the following questions.

- If $(x, y)$ is an edge in $G$ and $x$ has depth 2 in $T$, then $y$ may have depth __(30)__ in $T$.
(A) 0 (B) 1 (C) 2 (D) 3 (E) 4

- If nodes $x$ and $y$ both have depth 2 in $T$ and $x \neq y$, then the shortest path (in terms of the number of edges) that connects $x$ and $y$ may have __(31)__ edges.
  (A) 0 (B) 1 (C) 2 (D) 3 (E) 4
- If node $r$ (the root of $T$) has degree 5 in $G$, then $r$ may have degree __(32)__ in $T$.
  (A) 1 (B) 2 (C) 3 (D) 4 (E) 5

15. (8%) The knapsack problem can be defined as follows. You may assume that it takes $O(1)$ time to perform an arithmetic operation on a pair of integers in $[U^2]$ for some $U \geq 2^{32}$ where $[X] := \{0, 1, 2, \ldots, X-1\}$.

    Input: an integer $m$ and $n$ pairs of positive integers $(w_1, v_1), (w_2, v_2), \ldots, (w_n, v_n)$ where $m, n \in [U]$ and $w_i, v_i \in [U]$ for every $i \in \{1, 2, \ldots, n\}$.

    Goal: output a subset $S$ of $\{1, 2, \ldots, n\}$ so that

    $$\sum_{i \in S} w_i \leq m \text{ and } \sum_{i \in S} v_i \text{ is the largest possible.}$$

    - If $m = \Theta(n^2)$, then the above knapsack problem is known to be solvable in __(33)__ time (as best as possible).
      (A) $O(n)$ (B) $O(n \log n)$ (C) $O(n^2)$ (D) $O(nm)$ (E) none of the above
    - If $m = \Theta(n^2)$ and $w_i \in \{1\}$ for every $i \in \{1, 2, \ldots, n\}$, then the above knapsack problem is known to be solvable in __(34)__ time (as best as possible).
      (A) $O(n)$ (B) $O(n \log n)$ (C) $O(n^2)$ (D) $O(nm)$ (E) none of the above
    - If $m = \Theta(n^2)$ and $w_i \in \{1, 2\}$ for every $i \in \{1, 2, \ldots, n\}$, then the above knapsack problem is known to be solvable in __(35)__ time (as best as possible).
      (A) $O(n)$ (B) $O(n \log n)$ (C) $O(n^2)$ (D) $O(nm)$ (E) none of the above

16. †† (8%) A Hamiltonian cycle of a graph $G$ is a simple cycle that visits all the nodes in $G$. Suppose that there is an $O(n^7)$-time algorithm that decides HamC($G$) for any $n$-node graph $G$.

    HamC($G$)

    Input: a simple undirected graph $G$

    Output: "true," if $G$ has a Hamiltonian cycle; "false," otherwise.

    Complete Algorithm 5, which is an $O(n^7)$-time algorithm that uses HamC($G$) at most once to decide HamC3($G, x, y, z$) for any $n$-node graph $G$, for some distinct nodes $x, y, z \in G$.

    HamC3($G = (V, E), x, y, z$)

    Input: a simple undirected graph $G = (V, E)$ of $|V| \geq 3$ and three distinct nodes $x, y, z \in G$.

    Output: "true," if $G$ has a Hamiltonian cycle $C$ on which $x, y, z$ are consecutive nodes in an arbitrary order; "false," otherwise.

**Algorithm 5:** HamC3($G=(V,E),x,y,z$)

1 $U \leftarrow V \cup \{a,b\}$;
2 $F \leftarrow E$;
3 **if** *at least two of edges* $(x,y),(y,z),(z,x)$ *are not contained in* $E$ **then**
4     **return** ______;
5 **else if** *exactly one of edges* $(x,y),(y,z),(z,x)$ *is not contained in* $E$ **then**
    /* Assume w.l.o.g. that $(x,y) \notin E$ */
6     $F \leftarrow F \cup \{\_\_\_\_\_\_\}$;
7 **else**
8     $F \leftarrow F \cup \{\_\_\_\_\_\_\}$;
9 **end**
10 **return** HamC($H=(U,F)$);

- Which of the following shall be placed in the missing part ㊱ of Line 4?
(A) true (B) false (C) HamC($G=(V,E)$) (D) HamC($H=(U,F)$) (E) none of the above

- Which of the following shall be placed in the missing part ㊲ of Line 6?
(A) $(a,x),(b,y)$ (B) $(a,y),(b,z)$ (C) $(a,z),(b,x)$ (D) $(a,b)$ (E) none of the above

- Which of the following shall be placed in the missing part ㊳ of Line 8?
(A) $(a,x),(b,y)$ (B) $(a,y),(b,z)$ (C) $(a,z),(b,x)$ (D) $(a,b)$ (E) none of the above