SCC 121 Fundamentals of Computer Science 2023-2024

Term 1 Coursework

Deadline for submission: Wednesday December 13, 2023, at 20:00

Submission is electronic as **PDF** file on Moodle.

This coursework is worth 20% for both BJTU and LU credits.

Please answer all questions. You must do this coursework completely by yourself. In this instance, it means **you are not even allowed to discuss the coursework questions with other students**. Your submitted files will automatically be checked for plagiarism.

You are allowed to ask the instructor about these questions. But I can only clarify the questions or give you suggestions if you get stuck, and I will try not to tell you whether you have the correct answer (to be fair to all students).

E-MAIL: haiyangliu@bjtu.edu.cn

Coursework Term 1 SCC 121

For the following 3 questions, let A = {1, 2, 3, 5}, B = {1, 4, 5, 9} and C = {3, 6, 9, 10}.

Q1. What is $(A \cup B) \cap (B \cup C)$? [1 mark]

Q2. What is $(A \cap B) \cup (A \cap C)$? [1 mark]

Q3. What is A - B - C? [1 mark]

For the following 5 questions, let f(x) = 2x + 4 and g(x) = 3x - 1, h(x) = g(x) - f(x).

Q4. What is the function f + g? [1 mark]

Q5. What is the function f * h? [2 marks]

Q6. What is the function $g \circ h$? [2 marks]

Q7. Find the inverse function of g(x). [2 marks]

Q8. Define f(x) recursively. [2 marks]

Q9. Convert the following English statement into a proposition.

“It's snowing today, but the roads aren't slippery.”, where A = “It's snowing today.” and B = “Roads are slippery.”. [1 mark]

For the following 2 questions, here are some equivalences.

- Implication: $(P \rightarrow Q) \Leftrightarrow (\sim P \vee Q)$
- Associativity: $(P \vee Q) \vee R \Leftrightarrow P \vee (Q \vee R)$
- Commutativity: $(P \vee Q) \Leftrightarrow (Q \vee P)$
- Domination: $(P \vee True) \Leftrightarrow True$

Q10. To convert $(D \vee E) \vee \sim G$ into $\sim G \vee (D \vee E)$, which equivalence do you apply? [1 mark]

Q11. To convert $\sim(D \vee E) \rightarrow G$ into $D \vee E \vee G$, which equivalence do you apply? [2 marks]

Q12. What is the decimal equivalent of the two's complement 8-bit binary 1101 1110? [2 marks]

Q13. Convert the following binary number to hexadecimal. [2 marks]
1101 1011 0111 0001

Consider the following C code, used in Qs 14-16.

```
char play[] = "football";
strcpy(play, "basketball");
```

Q14. What is the size of the array `play`? [1 mark]

Q15. If the runtime allocates the character '`f`' in `play` at address 128 (decimal), what is the address of the character '`b`', assuming each byte has its own address?
[1 mark]

Q16. What is the result after executing statement `strcpy(play, "basketball")`? [2 marks]

Consider the following C code, used for Qs 17-18.

```
int x;
scanf("%d", &x);
int res = 0;
for (int i = 0; i < x; ++i) {
     if (i * i % x == 1) {
     ++res;
     }
}
printf("%d", res);
```

Q17. What is the output after executing this code if the input value is 15? [2 marks]

Q18. What is the time complexity of this code? [2 marks]

Q19. What is the time complexity of the following code? [2 marks]

```
int i, j, k = 0;
for (i = n / 2; i <= n; i++) {
    for (j = 2; j <= n; j = j * 2) {
        k = k + n / 2;
    }
}
```

A. O(n)
B. O(n log n)
C. $O(n^2)$
D. $O(n^2 \log n)$

Q20. Consider there is a linked list with each node defined as follows:

```
typedef struct Node {
     int data;
     struct Node* next;
}
```

There is a pointer to the head of the linked list: `Node* head`. If you want to insert a new node in the linked list whose member data has the value 42 and make the new node the first node of the linked list, write the code: [3 marks]

First line of the code is given:

```
Node* newNode = (Node*)malloc(sizeof(Node));
```

Consider the following C code, used for Qs 21-22.

```
int y = 0;
int *p;
int main(){
    int x;
    p = malloc(sizeof(x));
    return 0;
}
```

Q21. Which memory region do we use to store `y`? Which memory region do we use to store `x`?
[2 marks]

Q22. Which region is variable `p` stored? Which region does `p` point to? [2 marks]

Q23. What does it mean when we say that an algorithm X is asymptotically more efficient than Y?
[2 marks]

A.X will always be a better choice for small inputs.
B.X will always be a better choice for large inputs.
C.Y will always be a better choice for small inputs.
D.X will always be a better choice for all inputs.

Q24. The number of executions of the underlined statement in the following algorithm is:
[2 marks]

```
int m = 0, i, j;
for (i = 1; i <= n; i++)
   for (j = 1; j <= 2 * i; j++)
      m++;
```

Q25. Consider the code fragment about insertion sort below. Assume n is the size of the input array.

```
void insertionSort() {
      for (int i = 1; i <= n; i++) {
            int x = a[i];
            int j = i - 1;
            while (__________ && __________) {
                  a[j + 1] = a[j];
                  ______;
            }
            ____________;
      }
}
```

Fill in the blank to complete the insertion sort algorithm. [4 marks]

The following code fragment use the idea of binary search to solve the problem of deciding if a non-negative integer is a perfect square, used for Qs 26-27.

```
bool isPerfectSquare (int num){
      int hi = _____;
      int lo = _____;
      while(____________){
            int mid = (lo + hi)/2;
            if(_________________) return true;
            else if(_________________) lo = mid +1;
            else hi = mid - 1;
      }
      return false;
}
```

Q26. Fill in the blanks to finish the algorithm. [5 marks]

Note: In the requirements of this question, 0 is a perfect square number.

Q27.How many times will the code in while loop execute respectively if the input is 0, 1, 16, 20? [4 marks]

Q28.Complete the code as required.

In an online game there are n pieces of equipment, and wearing the *i*-th piece of equipment requires the player to have a strength value of at least a_i, and wearing that piece of equipment increases the player's strength value by b_i. Now asking what is the **minimum initial strength value** of the player to be able to wear all of the pieces of equipment in a certain order?

Input: The first line is an integer n, stating the number of the equipment; The second line contains n integers, the i-th integer a_i is the strength requirement for i-th equipment; The third line contains n integers, the i-th integer b_i is the increase strength value for the player when equipping the i-th equipment.

Output: A single integer, the minimum initial strength value.

Note: All data are sorted from smallest to largest, so the code uses binary search to find the minimum value that satisfies the condition.

```
#include <stdio.h>
const int maxn = 1005;
int n;
int a[maxn], b[maxn];

bool check(int x) {
    for (int i = 1; i <= n; i++) {
        if (    (1)    ) {
            x += b[i];
        } else {
            return false;
        }
    }
    return true;
}
int main() {
    scanf("%d", &n);
    for (int i = 1; i <= n; i++)
        scanf("%d", a + i);
    for (int i = 1; i <= n; i++)
        scanf("%d", b + i);
    int ans = 1145141919;
    int l = 1, r = ans, mid = (l + r) / 2;
    for (;  (2)  ; mid = (l + r) / 2) {
        if (check(mid)) {
            ans = mid;
            (3)    ;
        } else {
            (4)    ;
        }
    }
    printf("%d", ans);
    return 0;
}
```

The blank (1) should be filled in ( ). [2 marks]

A. `x >= a[i]` B. `x >= a[x]`

C. `x <= a[x]` D. `x <= a[i]`

The blank (2) should be filled in ( ). [2 marks]

A. `l < r` B. `l <= r`

C. `check(l)` D. `check(r)`

The blank (3) should be filled in ( ). [1 marks]

A. `l = mid + 1` B. `l = mid - 1`

C. `r = mid + 1` D. `r = mid - 1`

The blank (4) should be filled in ( ). [1 marks]

A. `l = mid + 1` B. `l = mid - 1`

C. `r = mid + 1` D. `r = mid - 1`