國 立 清 華 大 學 命 題 紙

96學年度 科技管理研究 系（所） 乙（資管） 組碩士班入學考試

科目 計算機概論 科目代碼 5101 共 8 頁第 1 頁 *請在【答案卷卡】內作答

Master Entrance Exam (2007)

- There are 10 questions in this exam. Each question is worth ***10 points***, with a total of ***100 points***.
- You may answer the exam in English, Chinese, or mixed.
- Good luck!

Question 1

Consider the following code segment.

```
int csi(int num[], int number)
{
int i, j, tmp, kount = 0;

for ( i = 0; i < (number - 1); i++)
{
for(j = 1; j < number; j++)
{
if (num[j] < num[j-1])
{
tmp = num[j];
num[j] = num[j-1];
num[j-1] = tmp;
kount++;
}
}
}

return kount;
}
```

(a) What is the worst-case complexity of this function (the Big-O notation)?

(b) What is the best-case complexity of this function (the Big-O notation)?

(c) Under what situation will the best-case complexity occur?

(d) What is the meaning of the variable `kount` ?

Question 2

The maximum contiguous subsequence sum problem can be defined as follows.

> *Given (possible negative) integers $A_1, A_2, \ldots, A_N$, find (and identify the sequence corresponding to) the maximum value of $\sum_{k=i}^{j} A_k$. The maximum contiguous subsequence sum is zero if all the integers are negative.*

For example, with the sequence of integers (4, -3, 5, -2, -1, 2, 6, -2), the maximum contiguous subsequence sum is 11 (from number 4 to number 6). Let us consider a divide-and-conquer algorithm to solve it. Divide the original sequence data into two halves, then the maximum contiguous subsequence sum can occur in one of three ways.

Case 1: It resides entirely in the first half.
Case 2: It resides entirely in the second half.
Case 3: It begins in the first half but ends in the second half.

Now you only know that Case 1 and Case 2 can be solved using recursive method. Case 3 can be solved in linear time. Therefore, we have a recursive maximum contiguous subsequence sum to solve this problem. This algorithm is implemented as following: (1) Perform linear work to compute a sum of two halves of original data sequence and them perform two recursive calls on each half to find the maximum contiguous subsequence sum in each half; (2) For each half of sequence data, do processes as in step (1).

Assume *T(N)* is the running time for this algorithm. *T(N)* can be constructed as:

$$T(N) = 2T(N/2) + O(N)$$

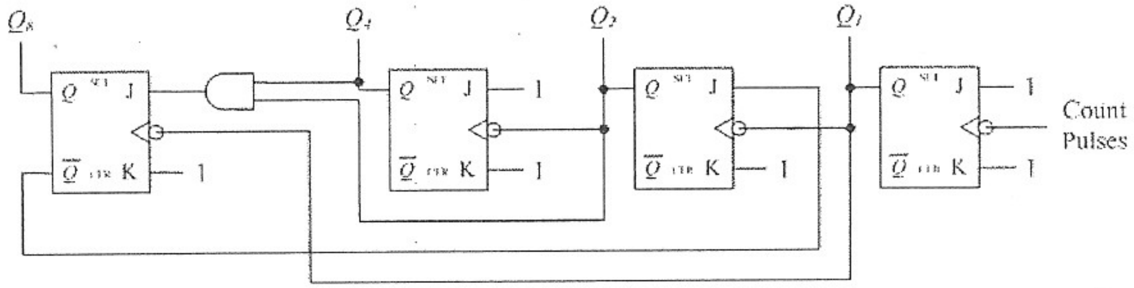
Assume *T(1)* is constant and *N* is a power of 2. Your job is to find the total running time (*Big-O time complexity*) for this algorithm using the information provided. Make your own assumptions when necessary. **You have to show calculation processes to get credits.**

Question 3

Consider a digital camera of 7 Mega Pixel. Each pixel has three primary colors, and each color has 64 intensities. This camera can take 5 pictures within one second. What is the data rate of this camera?

Question 4

Consider the following logic diagram of a ripple counter implemented by four *JK* flip-flops and an *AND* gate. Q_1, Q_2, Q_4, and Q_8 are four binary bit data. The flip-flops trigger on the negative edge, i.e., when the CP (Count Pulses) signal goes from 1 to 0.



Based on the about information, answer the following questions:

(a) Show the timing diagram of Q_1, Q_2, Q_3, and Q_4 corresponding to CP inputs.
(b) Show the state diagram of this ripple counter.

Question 5

For this phrase, “GO GO GO NTHU”
Please use Huffman Coding to encode each letter using the least number of bits.

Question 6

The Post Office Protocol (POP3) is a popular example of a pull based mail delivery protocol. In this protocol, email readers (or clients) polls the mail server to see if there is new mail, and if so downloads it to the users. In contrast, there is another type called push based mail delivery protocol. In this case, the client can become aware almost instantly of the existence of new messages.

Please describe the design differences in a pull based vs. a push based mail delivery protocol. (Your answer should focus on the *differences* in comparing the design considerations for the two protocols.)

Question 7

Consider when design a database system,

(a) what is concurrency control?

(b) The concept of "transaction" is introduced in a database system to ensure that the integrity of a database is maintained. In a SQL-like language, it would look like:

BEGIN the Transaction
EXECUTE *several queries*
COMMIT the Transaction

For this question, please *write down queries examples* to demonstrate that the "Transaction" concept can help preserve the integrity of a database. That is, in case part of the queries fails, with "Transaction" making several queries as a block, the database system can either rollback the entire transaction or just the failed query. You also need to point out the *possible incorrect results* when the queries examples are executed without using the "transaction" concept.

Question 8

(a) **Explain the following terms and their purposes.**

- i) **TCP**
- ii) **DHCP**
- iii) **NAT**
- iv) **Broadcast**
- v) **Autonomous System**

(b) Consider a datagram network using 32-bit host addresses. Suppose a router has four links, numbered 0 through 3, and packets are to be forwarded to the link interfaces as follows:

Destination Address Range	Link Interface
11100000 00000000 00000000 00000000 ~ 11100000 11111111 11111111 11111111	0
11100001 00000000 00000000 00000000 ~ 11100001 00000000 11111111 11111111	1
11100001 00000001 00000000 00000000 ~ 11100001 11111111 11111111 11111111	2
Otherwise	3

i) Provide a forwarding table that has four entries, uses longest-prefix matching, and forwards packets to the correct link interfaces.

ii) Describe how your forwarding table determines the appropriate link interface for datagrams with destination addresses:

```
11001000 10010001 01010001 01010101
11100001 00000000 11000011 00111100
11101110 10000000 00010001 01110111
```

Question 9

Given the following complete program:

```
#include <iostream>
#include <vector>
using namespace std;

void foo(unsigned int num, vector<bool> &vec);
void fun(const vector<bool> &vec);
void bar(vector<bool> &vec);
void morefun(const vector<bool> &vec);

int main()
{
   const unsigned int Number = 101;
   vector<bool> BitVec;
   foo(Number, BitVec);

   cout << "After foo BitVec = ";
   fun(BitVec);
   cout << endl << endl;
   bar(BitVec);
   cout << "After bar BitVec = ";
   fun(BitVec);
   cout << endl << endl;
   cout << "The number " << Number << " equals ";
   morefun(BitVec);
   cout << " in binary." << endl << endl;
   return 0;
}
```

```
   void foo(unsigned int num, vector<bool> &vec)
   {
      while (num != 0)
      {
         bool bit = ((num % 2) == 1);
         vec.push_back(bit);
         num = num / 2;
      }
   }

   void fun(const vector<bool> &vec)
   {
      for (unsigned int ind = 0; ind < vec.size(); ++ind)
      if (vec[ind])
         cout << '1';
      else
         cout << '0';
   }

   void bar(vector<bool> &vec)
   {
      unsigned int size = vec.size();
      for (unsigned int ind = 0; ind < size / 2; ++ind)
      {
      bool bit = vec[ind];
      vec[ind] = vec[(size - 1) - ind];
      vec[(size - 1) - ind] = bit;
   }
}

void morefun(const vector<bool> &vec)
{
   typedef vector<bool>::const_iterator VecIt;
   const VecIt end = vec.end();

   for (VecIt it = vec.begin(); it != end; ++it)
      if (*it)
         cout << '1';
      else
         cout << '0';
}
```

Determine the output that is produced when the program is run.

Question 10

Consider a system consisting of ***m*** resources of the same type, being shared by ***n*** processes. Resources can be requested and released by processes only one at a time. Describe and prove the required conditions that the system is deadlock free.