浙江大学 2011 - 2012 学年秋学期

《数据结构基础》课程期末考试试卷(A)

课程号： 211C0020 ，开课学院： 计算机科学与技术

考试试卷： √A 卷、B 卷（请在选定项上打 √）

考试形式： √闭、开卷（请在选定项上打 √），允许带____无___入场

考试日期： 2011 年 11 月 7 日,考试时间： 120 分钟

诚信考试，沉着应考，杜绝违纪。

考生姓名：_____学号：_____所属院系：_____

题序	一	二	三	四	总 分
得分					
评卷人					

Answer Sheet

Part I (20)				
1. a	2. b	3. c	4. b	5. d
6. a	7. b	8. b	9. d	10. b

Part II (24)

1. ① `temp_width++` ③ `temp_width=0`

② `if ( p->rchild != NULL ) enqueue ( p->rchild );`

2. ① `Cvw < T[ W ].Dist` ③ `V`

② `Cvw`

3. ① `i < j` ② `Swap( &A[ i ], &A[ Right - 1 ] )`

Part III (41)

1. It still takes O(logN) time. The reduced problem size is either 0.99N or 0.01N, depending on which side is pruned. So $T(N) \le T(0.99N)+c$, where c is a constant. Repeatedly applying this to the reduced problem size, we have

$T(N) \le T(0.99N)+c \le 0.99^2N+2c \le \ldots \le 0.99^kN+kc$.

Let $0.99^kN=1$. $k\log(99/100) + \log N=0$. $k=-\log N/\log(99/100)= \log N/\log(100/99)$. Therefore, $T(N) \le \log N/\log(100/99)c=O(\log N)$.

2. (a) PPPOOPOOPPOOPO

(b) 2M-1.
First, every operand will be pushed once (hence M pushes).
Secondly, for every two top operands, the result of computation with the corresponding operator will be pushed back onto the stack as a new operand.
In an expression with M operands, there must be M-1 operators (assume binary operators only).
Therefore the total number of pushes is M+M-1.

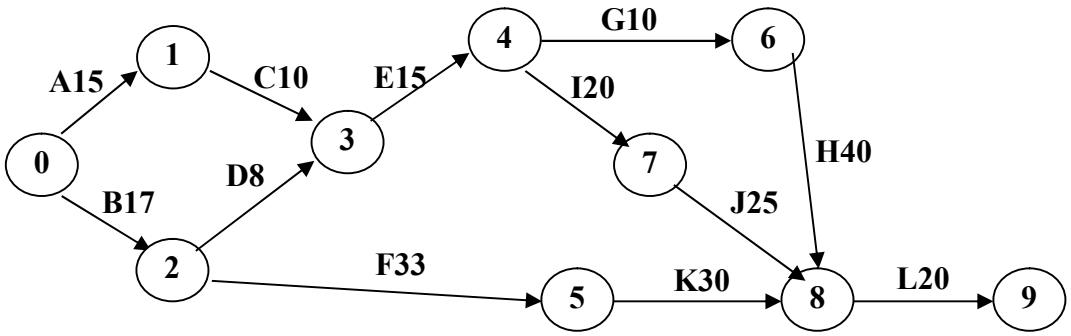
3. Union(1,2) -2 1 -1 -1 -1 -1 -1 -1 -1 –1
Union(3,4) -2 1 -2 3 -1 -1 -1 -1 -1 –1
Union(3,5) -2 1 -3 3 3 -1 -1 -1 -1 –1
Union(1,7) -3 1 -3 3 3 -1 1 -1 -1 –1
Union(3,6) -3 1 -4 3 3 3 1 -1 -1 –1
Union(8,9) -3 1 -4 3 3 3 1 -2 8 –1
Union(1,8) -5 1 -4 3 3 3 1 1 8 –1
Union(1,3) -9 1 1 3 3 3 1 1 8 –1
Union(9,10) -10 1 1 3 3 3 1 1 1 1

4.

i	0	1	2	3	4	5	6	7	8	9	10	11	12
H[i]	161	92		81		57	58		21		45	117	38

5.

(a)



(b) 110

(c) ACEGHL & BDEGHL

Part IV (15)

```
Push ( Stack S, int key ) ;
int Pop( Stack S ) ;
int PeekMedian( Stack S ) ;

Use quick-select to find the Kth largest key :
0. use another integer array for selection;
1. if Size = 1, return the element;
2. Pick a pivot;
3. Partition by the pivot ; - O(Size)
4. if (K<=|left|) quickselect(K, left) ;
5. if (K==|left|+1) return pivot ;
6. else quickselect(K-|left|-1, right) ;

Push = Push+insert – O(1).
Pop = Pop+Find+Delete - O(Size).
PeekMedian = worst O(Size^2), average O(Size).

With N operations :
Push – O(N)
Pop – worst O(N^2)
PeekMedian – worst O(N^3), average O(N^2)

Extra space = O(N)
```

NOTE: Please write your answers on the answer sheet.
注意：请将答案填写在答题纸上。

I. Please select the answer for the following problems. (20 points)

(1) Which of the following functions grows the fastest for sufficient large N?

a. $N^{1-\frac{1}{1000}}$ b. $N \log N$ c. $\log^2 N$ d. $1000 \log N$

(2) Which of the following relations is true?

a. $N = O(\log^{1000} N)$ b. $N \log \log N = O(N \log^2 N)$

c. $N \log N = O(N \log \log N)$ d. $N \log N^2 = \Omega(N \log^2 N)$

(3) There is a kind of queue that could insert object into both ends and delete object in only one end. After inserting a, b, c, d, e in order into this initial empty queue, which of the followings is the ***impossible*** output?

a. b,a,c,d,e b. d,b,a,c,e c. d,b,c,a,e d. e,c,b,a,d

(4) Given two traversal sequences produced by preorder and postorder traversing a binary tree respectively. If the two traversal sequences have exactly the opposite order, which form does the tree certainly have? (Note: *height* is defined to be the number of edges along the longest path from the root.)

a. NULL or has only one node
b. height = number of nodes - 1
c. any node in the tree has no left branch
d. any node in the tree has no right branch

(5) There are 8 leaf nodes on the sixth level of a complete binary tree. Suppose that the root is on the first level, then how many nodes does the complete binary tree have at most?

a. 39 b. 52 c. 79 d. 111

(6) Comparing the extra space required to sort a list of elements, the proper relations among the heapsort, quicksort and mergesort is ____.

a. heapsort < quicksort < mergesort
b. heapsort > quicksort > mergesort
c. heapsort < mergesort < quicksort
d. heapsort > mergesort > quicksort

(7) Which of the following array is the result of inserting 6, 4, 3, 5, 8, 9 in order into an initially empty maxheap?

a. 9,8,6,5,4,3 b. 9,6,8,4,5,3 c. 9,6,8,3,4,5 d. 6,4,8,3,5,9

(8) If quadratic probing is used and there are spaces in the table, and the table size is prime, then which one of the following is true?

a. A new element can always be inserted
b. A new element may not be inserted
c. A new element can not be inserted
d. None of the above

(9) Place M items in a hash table with an array size of S, the loading factor is _____.

a. S+M b. M-S c. M×S d. M/S

(10) Given a weighted and connected undirected graph G, of which some edges have the same weight. Then there is/are _____ minimum spanning tree(s) of G.

a. only one b. one or more
c. more than one d. zero or more

II. Given the function descriptions of the following two (pseudo-code) programs, please fill in the blank lines. (24 points)

(1) The function is to find the width of a binary tree T by levelorder traversal. The width of a binary tree is the maximum number of nodes on one level of the tree. The functions queue_rear() and queue_front() returns the current rear and front positions in the queue, respectively. (9 points)

```
int Width( BinTree T )
{     BinTree  p;
      int  Last = queue_rear();  /* Last points to the last node on a level */
      int  temp_width=0,  max_width=0;
      if ( T == NULL) return 0;
      else {
            enqueue ( T );
            while (queue is not empty) {
                  p = dequeue( );
                  ①_____;
                  if ( p->lchild != NULL )  enqueue ( p->lchild );
                  ②_____;
                  if ( queue_front() > Last ) {
                        Last = queue_rear();
                        if ( temp_width > max_width ) max_width = temp_width;
                        ③_____;
                  } /* end-if */
            } /* end-while */
            return  max_width;
      } /* end-else */
}
```

(2) The function is to use Prim method finding the minimum spanning tree of a given graph G. Here Table T is an array which records the distance, T[i].Dist, from i to the current known spanning tree, and the parent, T[i].Parent, of i. Vertex Start is the root of the spanning tree. InitTable(Start, G, T) initializes all the T[i].Dist to be Infinity, and T[i].Parent to be NotAVertex except that T[Start].Parent is -1. Assume that the weight of the edge between V and W is denoted by C_{VW}. (9 points)

```
void Prim( Vertex Start, Graph G, Table T )
{    Vertex  V, W;
     InitTable(Start, G, T);
     V = Start;
     while( V is a legitimate vertex ) {
        T[ V ].Dist = 0;   /* V is included in the current known spanning tree */
        for ( each W adjacent to V )
               if ( T[ W ].Dist != 0 )
                  if ( ①_____ ) {
                        T[ W ].Dist = ②_____;
                        T[ W ].Parent = ③_____;
                  } /* end-if update W */
        V = smallest non-zero distance vertex;
       } /* end-while */
}
```

(3) The function is quick sort with median of three. (6 points)

```
void  Qsort( ElementType A[ ], int Left, int Right )
{   int  i,  j;
    ElementType  Pivot;
    if ( Left + Cutoff <= Right ) {
        Pivot = Median3( A, Left, Right );
        i = Left;
        j = Right – 1;
        for( ; ; ) {
             while ( A[ + +i ] < Pivot ) { }
             while ( A[ – –j ] > Pivot ) { }
             if ( ①_____ )
                    Swap( &A[ i ], &A[ j ] );
             else  break;
        } /* end-for */
        ②_____;
        Qsort( A, Left, i - 1 );
        Qsort( A, i + 1, Right );
    } /* end-if */
    else  InsertionSort( A + Left, Right - Left + 1 );
}
```

III. Please write or draw your answers for the following problems on the answer sheet. (41 points)

(1) A balanced binary search divides the sequence into 50%:50% equal sized subsequences. Now we define the unbalanced binary search to be always dividing the sequence into 1%:99% subsequences. That is, we always test at the position corresponding to the fraction 1/100 of the input of size N (e.g., if N=1000, we will test the position 10 of the array). Does the binary search still take O(logN) time? Please justify your answer. (6 points)

(2) A stack is used to evaluate a postfix expression. Let P stand for "push" and O for "pop", then
   (a) what is the sequence of stack operations for evaluating the postfix expression 1 2 3 + * 4 - ? (4 points)
   (b) In general, when a postfix expression with binary operators only consists of M operands, how many push (P) operations will be taken during the expression evaluation? Please justify your answer. (4 points)

(3) The array representation of the disjoint sets is given by S[] with S[i] being initialized to be -1 for all i. Please list the resulting array elements after invoking: union(1,2), union(3,4), union(3,5), union(1,7), union(3,6), union(8,9), union(1,8), union(1,3), and union(9,10). Note: Assume union-by-size and find-with-path-compression, and the elements are numbered from 1 to 10. (5 points)

(4) Given a hash table of size 13 and the hash function $H(key) = key \bmod 13$. Assume that double hashing is used to solve collisions with $H(key) = (H(key) + i*H_2(key)) \bmod 13$ for $i = 1,2,\ldots,12$, where $H_2(key) = (key \bmod 11) + 1$. Please fill in the hash table with input numbers { 92,81,58,21,57,45,161,38,117 }. (9 points)

(5) A project consists of 12 activities named from A to L whose lasting times and prerequisites are listed in the following table. Please
   (a) draw the corresponding AOE (Activity On Edge) network (6 points);
   (b) evaluate the time that is required to finish the project (3 points); and
   (c) point out the critical path(s) (4 points);

Activity	Lasting time	Prerequisites
A	15	none
B	17	none
C	10	A
D	8	B
E	15	C, D
F	33	B

Activity	Lasting time	Prerequisites
G	10	E
H	40	G
I	20	E
J	25	I
K	30	F
L	20	H, J, K

IV. The basic operations of a stack are Push (inserting an element into the top of the stack) and Pop (deleting the top element of the stack with its value returned). Now suppose that we need another operation PeekMedian – that is to return the median value (the Size/2-th largest value) of all the elements in the stack. Please describe the algorithms you use to implement Push, Pop and PeekMedian for such a stack with integer elements. Assume that the number of each operation will be N. You must obtain the best overall performance of all the operations. Please discuss the time complexity and extra space (if any) complexity of your algorithms. (15 points)