

Domanda 1

Risposta non data

Punteggio max.: 2,00

Given the following sequence of pairs, where the relation $i-j$ means that node i is adjacent to node j , apply an on-line connectivity algorithm with quick-union. Node names are integers in the range from 0 to 9.

9-3 0-9 7-6 9-1 6-4 2-7 8-3 4-8 2-9 8-4 0-5 7-6

Show the content of the array at the end of the process.

Risposta:



La risposta corretta è : 3 5 4 1 1 5 4 6 1 3

Domanda 2

Risposta non data

Punteggio max.: 1,00

Perform the first 2 steps of quicksort to sort the following array of integers in ascending order. Steps must be improperly considered in breadth on the recursion tree, rather than in depth.

16 3 14 11 12 5 2 9 8 11 6 13 4 7 9

Which is the value stored in element [5] after these 2 steps (e.g, 2)?

Risposta: ✖

La risposta corretta è : 5

Domanda 3

Risposta non data

Punteggio max.: 1,00

Perform the first 2 steps of quicksort to sort the following array of integers in ascending order. Steps must be improperly considered in breadth on the recursion tree, rather than in depth.

16 3 14 11 12 5 2 9 8 11 6 13 4 7 9

Which is the value stored in element [12] after these 2 steps (e.g, 2)?

Risposta: ✖

La risposta corretta è : 14

Domanda 4

Completo

Punteggio ottenuto 0,00 su 2,00

Analyze the following function.

```
int f (int *v, int l, int r, int k) {  
    int c;  
    if (l > r) return (-1);  
    c = (l+r) / 2;  
    if (k < v[c]) return (f (v, l, c-1, k));  
    if (k > v[c]) return (f (v, c+1, r, k));  
    return c;  
}
```

Write the recurrence equation describing the algorithm.

Commento:

Missing.

Domanda 5

Risposta non data

Punteggio max.: 2,00

Given a binary tree, its visits return the following sequences.

Pre-order L M O U K X I H T Y Z S P

In-order O M L K X U H T Y I Z P S

Post-order O M X K Y T H P S Z I U L

Which are the leaves of the tree? Report them from left to right (e.g., A B C etc.).

Risposta:



La risposta corretta è : O X Y P

Domanda 6

Risposta non data

Punteggio max.: 2,00

Suppose to have an initially empty priority queue implemented on an array as a minimum heap. Consider Given the following sequence of integers and "*" characters, where each integer corresponds to an insertion into the priority queue and each character "*" corresponds to an extraction.

3 9 11 13 4 19 10 15 7 1 * *

Report the sequence of values stored in the array representing the priority queue at the end of the entire process (e.g., 23 12 16 etc.).

Risposta:



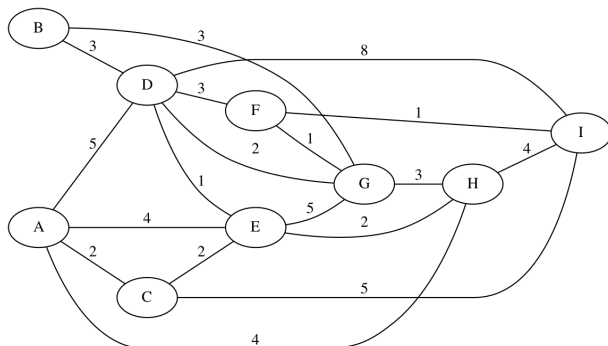
La risposta corretta è : 4 7 10 13 9 19 11 15

Domanda 7

Risposta non data

Punteggio max.: 1,00

Given the following undirected and weighted graph find a minimum spanning tree using Prim algorithm starting from vertex A.



Which is the weight of the minimum weight as a result (e.g., 34).

Risposta:



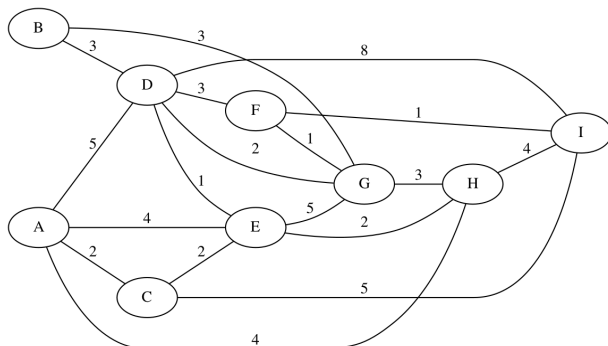
La risposta corretta è : 14

Domanda 8

Risposta non data

Punteggio max.: 1,00

Given the following undirected and weighted graph find a minimum spanning tree using Prim algorithm starting from vertex A.



Specify the list of the weights of the selected edges in increasing order (e.g., 2 3 3 4 5 ...).

Risposta:



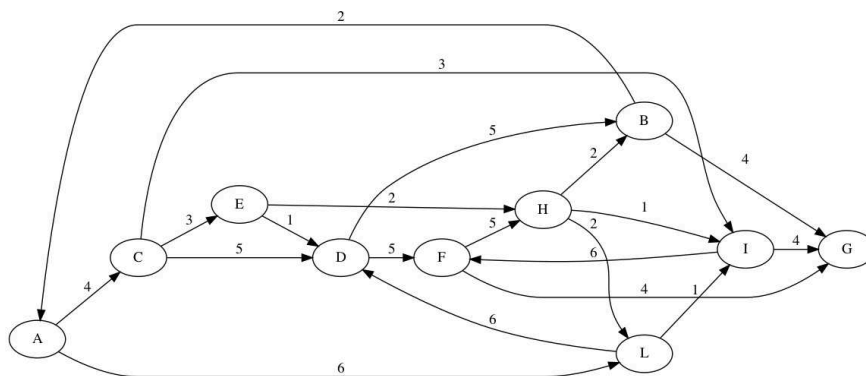
La risposta corretta è : 1 1 1 2 2 2 2 3

Domanda 9

Risposta non data

Punteggio max.: 2,00

Given the following directed and weighted graph, apply Dijkstra's algorithm to find all shortest paths connecting node A with all the other nodes. If necessary, consider nodes and edges in alphabetical order.



Specify the distance of all nodes in increasing order (e.g., 0 2 3 3 4 5 etc.).

Risposta:



La risposta corretta è : 0 4 6 7 7 8 9 11 11 13

Domanda 10

Risposta non data

Punteggio max.: 1,00

Analyze the following program.

```
#include <stdio.h>

int f1 (int);
int f2 (int);

int f1 (int n) {
    if (n<=0) {
        return n;
    }
    f2 (n-2);
    return n-2;
}

int f2 (int n) {
    if (n<=0) {
        return n;
    }
    f1 (n+1);
    return n+1;
}

int main () {
    fprintf (stdout, "%d\n", f1(6));
}
```

How many times f1 is called (e.g., 2)?

Domanda 11

Risposta non data

Punteggio max.: 1,00

Analyze the following program.

```
#include <stdio.h>

int f1 (int);
int f2 (int);

int f1 (int n) {
    if (n<=0) {
        return n;
    }
    f2 (n-2);
    return n-2;
}

int f2 (int n) {
    if (n<=0) {
        return n;
    }
    f1 (n+1);
    return n+1;
}

int main () {
    fprintf (stdout, "%d\n", f1(6));
}
```

Which is the value display by the main program (e.g., 6)?

Domanda 12

Risposta non data

Punteggio max.: 0,50

Analyze the following program.

```
#include <stdio.h>

int f1 (int);
int f2 (int);

int f1 (int n) {
    if (n<=0) {
        return n;
    }
    f2 (n-2);
    return n-2;
}

int f2 (int n) {
    if (n<=0) {
        return n;
    }
    f1 (n+1);
    return n+1;
}

int main () {
    fprintf (stdout, "%d\n", f1(6));
}
```

How many times f2 is called (e.g., 5)?

Domanda 13

Risposta non data

Punteggio max.: 3,00

The following function receives as input parameter the string s="This 12345 is a string".

```
void f (char *s) {  
    void f (char *s) {  
        int i, j;  
  
        i = 0;  
        while (i < strlen(s)) {  
            if (s[i]==' ' || (s[i]>='0' && s[i]<='9')) {  
                for (j=i+1; j < strlen(s)+1; j++) s[j-1] = s[j];  
            } else {  
                i = i + 1;  
            }  
        }  
  
        return;  
    }  
}
```

Which is the string stored in the parameter s when the function ends (e.g, ABC)?

Domanda 14

Completo

Punteggio ottenuto 0,00 su 5,00

A double linked list stores the data related to a set of students. Each element of the list (of type `list_t`) has a left pointer connecting it to the element on its left and a right pointer connecting it to the element on its right. Moreover, each element stores the last name, first name, and the number of examination passed by the student. Sort the list in ascending order using selection sort. At the end of the procedure the list must be sorted with the last name as first key, the first name as secondary key, and the number of exams passed as tertiary key. Write the function `sort_list` which receives and return the pointer to the head of the list. The C structure for the list and the function prototypes follow.

```
typedef struct list_s list_t;
struct list_s {
    char *last;
    char *first;
    int n;
    list_t *left, *right;
};

void sort_list (list_t *head);
```

Commento:
Missing.

Domanda 15

Risposta non data

Punteggio max.: 2,50

The following function must generate all anagrams of the string word, which do not start with a capital letter. While building the anagrams, the function stores characters into the string anagram. This string is displayed when an anagram is built that does not start with a capital letter. The array used of integer values includes a flag for each letter of the string anagram and it is initialized to all zeros.

```
void generate_r (char *word, char *anagram, int *used, int n) {  
    int i;  
  
    if (n == strlen (word)) {  
        // Insert code here: Position 1  
        ...  
    }  
  
    for (i=0; i < strlen(word); i++) {  
        if (used[i] == 0) {  
            // Insert code here: Position 2  
            ...  
        }  
    }  
  
    return;  
}
```

Complete the function by inserting the proper lines of code in position 1 and 2.

Domanda 16

Risposta non data

Punteggio max.: 7,00

A robot can make steps only of pre-defined lengths (e.g., 20, 40, 60 centimeters), moving only in one direction. Given the array l , its size n and a distance d that the robot must cover (e.g., 10500 centimeters), write the recursive function `cover` that finds all sequences of steps that makes the robot cover the specified distance.

For example if $l = \{2, 6\}$, $n = 2$, and $d = 10$, the function must find all possible solutions to cover 10 centimeters using steps of length 2 and 6 centimeters. Thus, there will be 4 solutions: $\{2, 2, 2, 2\}$, $\{2, 2, 6\}$, $\{2, 6, 2\}$, and $\{6, 2, 2\}$.

```
void cover (int *l, int n, int d);
```