

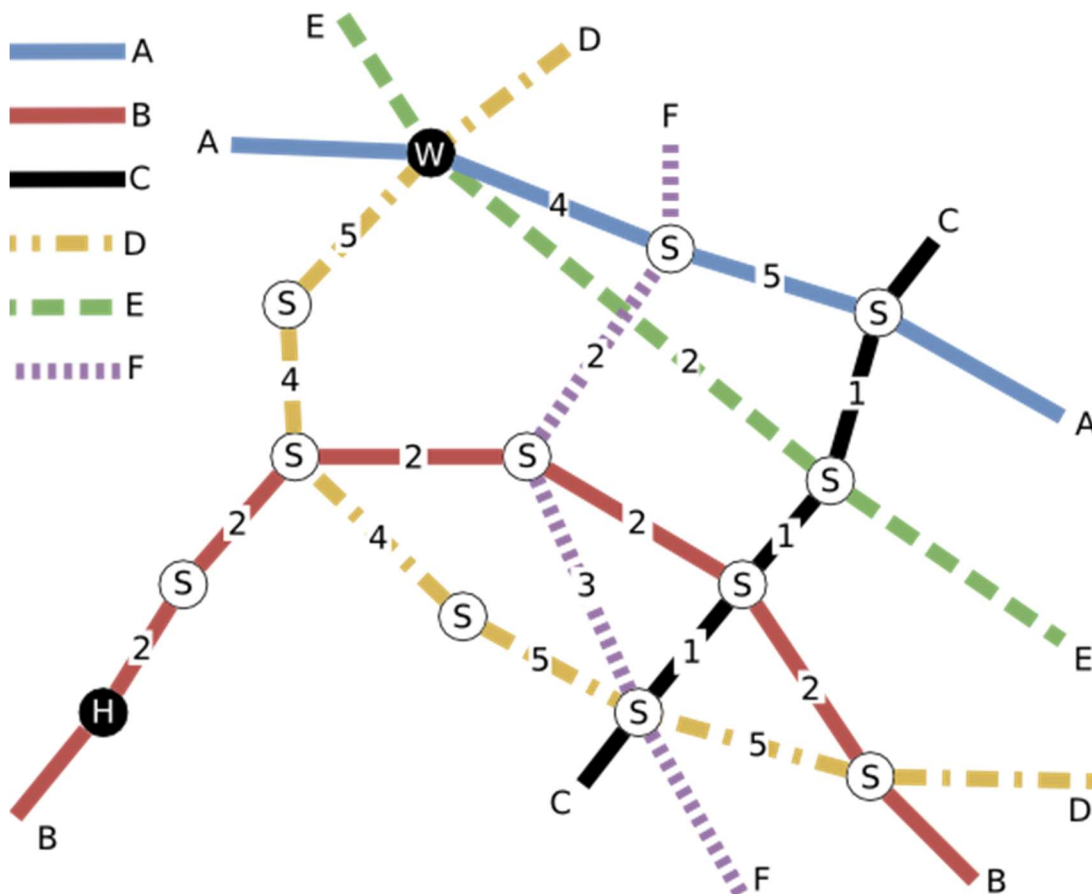
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
FIRST SEMESTER B.E CSE
COMPUTATIONAL THINKING MIDTERM EXAM

Marks : 25

TIME: 1 1/2 HOUR

- Jonathan goes to work by train every day. There is no direct line, so Jonathan has to switch between several lines. The map below shows the available lines with the travel time between any two stations (Jonathan's home is marked with "H", his workplace is marked with "W", and the stations, where it is possible to change line, are marked with "S").

Assuming that changing line takes no time, which lines should Jonathan take in order to arrive at work as fast as possible and write pseudocode for finding the fastest route? (10)



2. A group of friends, two women (Alice and Babs), and two men (Zach and Yabu), like to pair up to go out on dates to cool restaurants. There are four combinations they date in (Alice-Zach, Alice-Yabu, Babs-Zach and Babs-Yabu). The favourite restaurant of one of the men and one of the women is a place called Quonk. However, if those two eat together they always try new restaurants as do the other pair if together. Therefore, when exactly one, and only one, of the particular man and woman in question is on a date they eat at Quonk. When Alice goes out with Zach they go to Quonk.

Which, if any, other pair from those below eat at Quonk:

- 1) Alice and Yabu,
- 2) Babs and Zach,
- 3) Babs and Yabu, or
- 4) none of the other pairs eat at Quonk?

Explain the procedure of how you got the correct answer. (10)

3. Debug and highlight the errors in the following code. (5)

```
/*
 * C Program to Input Few Numbers & Perform Merge Sort on them using Recursion
 */

#include <stdio.h>

void mergeSort(int [], int, int, int)
void partition(int [],int, int)

int main()
{
    int list[50];
    int i, size;

    printf("Enter total number of elements:");
    scanf("%d", size);
    printf("Enter the elements:\n");
    for(i = 0; i < size1; i++)
    {
        scanf("%d", list[i]);
    }
    partition(list, 0, size - 1);
    printf("After merge sort:\n");
    for(i = 0;i < size; i++)
    {
```

```

    printf("%d " list[i]);
}

return 0;
}

void partition(int list[],int low)
{
    int mid;

    if(low < high);
    {
        mid = (low + high) / 2;
        partition(list, low, mid);
        partition(list, mid + 1, high);
        mergeSort(list, low, high);
    }
}

void mergeSort(int list[],int low,int mid,int high)
{
    int i, mi, k, lo, temp[50];

    lo = low;
    i = low;
    mi = mid + 1;
    while ((lo <= mid) && (mi <= high))
    {
        if (list[lo] <= list[mi])
        {
            temp[i] = list[lo];
            lo++;
        }
        else
        {
            temp[i] = list[mi];
            mi++;
        }
        i++;
    }
    if (lo > mid)
    {
        for (k = mi; k <= high; k++)
        {
            temp[i] = list[k];
            i++;
        }
    }
}

```

```
    }  
  }  
  else  
  {  
    for (k = lo; k <= mid; k++)  
    {  
      temp[i] = list[k];  
      i++;  
    }  
  }  
  
  for (k = low; k <= high; k++);  
  {  
    list[k] = temp[k];  
  }  
}
```