

Group Exercise #5

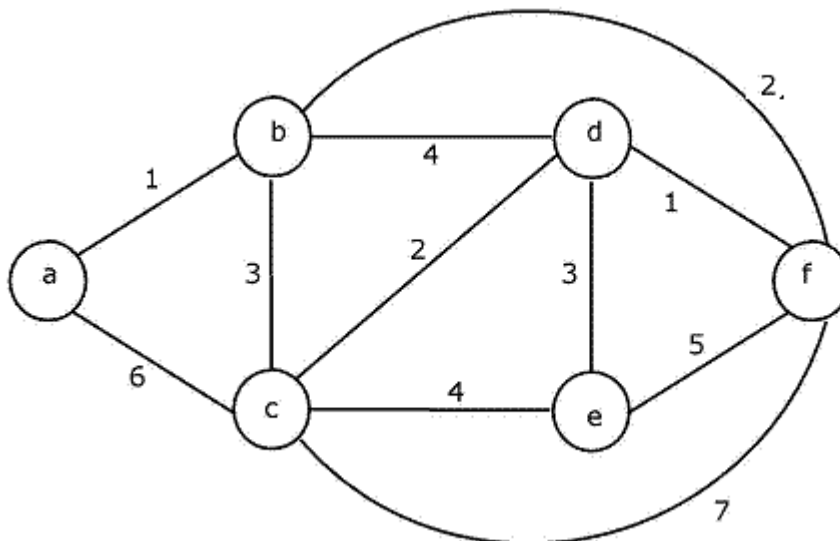
Due 11:59PM, Monday April 26

Instructions:

- 1) Find which group you are in from the following list (it is re-randomized since the last group).
 - a. Group 1 – Talha, Cale, ANDY
 - b. Group 2 – Tuva, Eddie, Oksana
 - c. Group 3 - Alejandra, Kevin, Kyleigh
 - d. Group 4 – David S, Malachi, Hayden
 - e. Group 5 – Garrett, David M, Jonathan
 - f. Group 6 – Alora, Jacobo, Luke
 - g. Group 7 – Aiden, Megan, Liam
 - h. Group 8 – Keith, Jonah, Nicole
 - i. Group 9 – Nemed, Marshall
- 2) Go to the discord server, introduce yourself in the channel for your group, and work out among your group who will work on which questions. Note that all channels are public.
- 3) Before the deadline discuss answers for each question in the group until there is consensus.
- 4) Create a PDF or Word Document with written/typed answers (**you do not need to make a video unless you want to**) and upload to your channel.

Questions

1. Given the following graph, find a minimum spanning tree using either Kruskal or Prim's algorithm, showing the steps along the way to generate the final MST.



2. For each of the following algorithms identify a recurrence relation or find the runtime in Theta/Big-O notation if there is no recurrence relation. **Don't solve the recurrence relations.**

a. Binary Search

```
int binary_search(int A[], int key, int imin, int imax)
{
    if (imax < imin):
        return KEY_NOT_FOUND;
    else
    {
        int imid = (imin + imax) / 2;

        if (A[imid] > key):
            return binary_search(A, key, imin, imid-1);
        else if (A[imid] < key):
            return binary_search(A, key, imid+1, imax);
        else
            return imid;
    }
}
```

b. Gaussian Elimination

```
for (int i = 0; i < N-1; i++)
{
    for (int j = i; j < N; j++)
    {
        double ratio = A[j][i]/A[i][i];
        for (int k = i; k < N; k++)
        {
            A[j][k] -= (ratio*A[i][k]);
            b[j] -= (ratio*b[i]);
        }
    }
}
```

c. Made-up algorithm

```
int arrayOdd(int A[], int n)
{
    if(n < 1)
        return 0;
    else
        return A[n-1]%2 + arrayOdd( A, n-1);
}
```

3. Solve this recurrence relation using the iterative substitution technique.

$$T(n) = T(n/2) + n$$