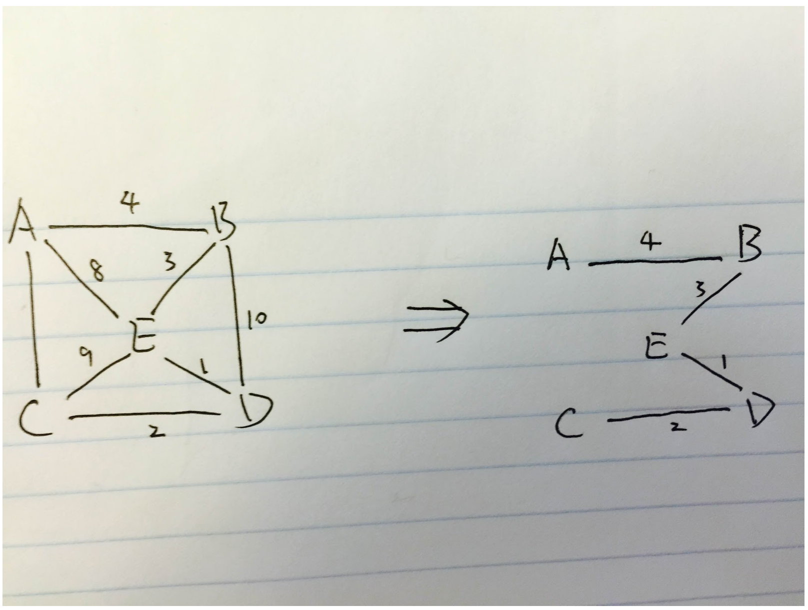
## CAS CS 330. Problem Set 6 (midterm-themed)

---

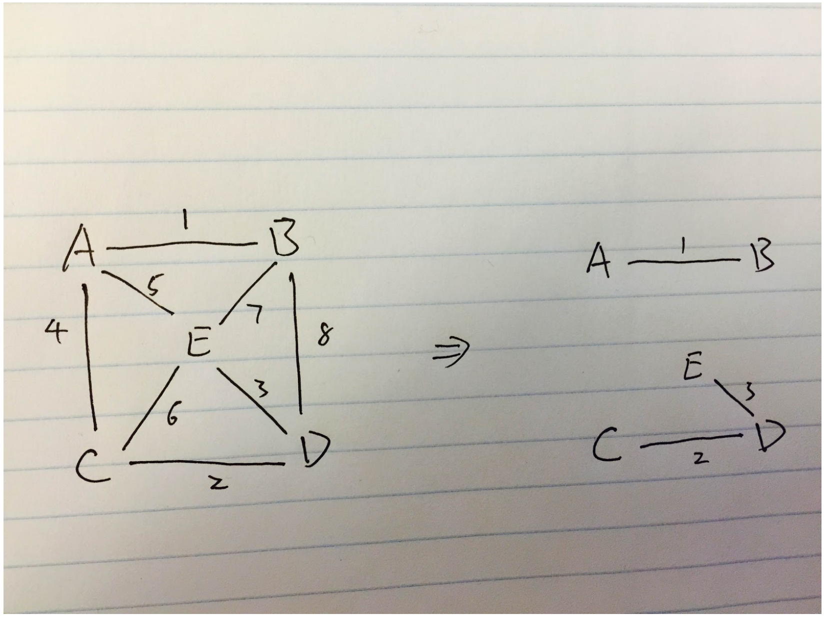
## Problem 1

**Proof:** Because this algorithm only looking for the shortest path for each node, this might causing a problem that this algorithm will, instead, finding two or more smaller spanning trees. This algorithm might ignore the important connecting path for two spanning trees because this path's weight is too big, and this algorithm decides to ignore this path. Therefore, this algorithm only gives us a subset of the minimum spanning tree.

Example 1: Algorithm provides with correct MST




Example 2: Algorithm doesn't work




## Problem 2

**Answer**: True

**Proof:** Because every boy is going to go through their list and propose to their dream girls one by one. Therefore, all the “dream girls” will be proposed before any other non-dreamgirls. Also, the number of dreamboys is larger than the number of dreamgirls,that is k<m. For the k+1th dream boy, all the dreamgirls have been proposed.

## Problem 3

**Algorithm:** We can use depth-first algorithm to find the path that go through minimum intersections. For example, we start at vertex K and go to vertex B.

```
PathList = [Number of the Nodes]
result = [ ]
Minimum(A)          {
        if Node A is B(our destination):
                return (B,0)
        if A is empty:
                return (A,0)
        for Vertex X that is connected to A:
                (Leaf, #OfNode)= Minimum(X)
                path = X + Leaf
                #OfNode = #OfNode + 1
```

we store the path and #OfNode only if the path contain the destination B

After we look through all the Vertices that connected to A, we find the path that go through minimum number of Node and return (path , #OfNode)

**Analyze:** This algorithm basically go through every vertex once, then updates and returns the value we need. So the best case and worst case is the same, both in Theta(n).

---

## Problem 4

**Algorithm:** for every coordinate in the input list, we pick up the first one (smallest), and set a cell phone base station at location on the west side of the house, which is four miles away.
then delete all the coordinates of houses that is in the range of this cell phone base station. Then do this algorithm again until the list is empty.

**Proof:** Apply this algorithm we will be able to set up bases stations that are not overlapped, and all the houses on the road get signal from one cellphone base station. So this algorithm would work.

**Analyze:** this algorithm goes through the elements on the list once, and every time it deletes the coordinates that is in the coverage of one cell phone station. Because all the coordinates are already sorted, we don't need to go through all the coordinates again, we can simply find the coordinates that is less them the coverage of the cellphone station. Therefore the final running time would be in Theta(n)

---

## Problem 5

A(x) = $a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x^1 + a_0$

step 1: evaluated on 4 points

$A_{odd} = a_7x^3 + a_5x^2 + a_3x^1 + a_1$

$A_{even} = a_6x^3 + a_4x^2 + a_2x^1 + a_0$

A(x) = $A_{even}(x^2)$ + x* $A_{odd}(x^2)$

step 2: evaluated on 2 points

$A_{odd1} = a_7x^1 + a_3$

$A_{0dd2} = a_5x^1 + a_1$

$$A_{even1} = a_6 x^1 + a_2$$
$$A_{even2} = a_4 x^1 + a_0$$

step 3 : evaluated on 1points

$$A_{odd11} = a_7$$
$$A_{odd12} = a_3$$
$$A_{odd21} = a_5$$
$$A_{odd22} = a_1$$
$$A_{even11} = a_6$$
$$A_{even12} = a_2$$
$$A_{even21} = a_4$$
$$A_{even22} = a_0$$

---

## Problem 6

(a). T(n) = 8T(n/2) + $n^2$

a = 8, b = 2, f(n) = $n^2$ then $n^{log_b^a} = n^3 >$ f(n)

according to master theory:

T(n) = Theta(f(n)) = Theta( $n^3$ )

(b).T(n) = 7T(n/2) + $n^2$

a = 7, b = 2, f(n) = $n^2$ then $n^{log_b^a} = n^{log_2^7}$ > f(n)

T(n) = Theta( $n^{log_2^7}$ )

(c).T(n) = 4T(n/2) + $n^2$

a = 4, b = 2, f(n) = $n^2$ then $n^{log_b^a} = n^{log_2^4}$ = f(n)

T(n) = Theta( $n^2 logn$ )

(d). T(n) = 2T(n/2) + $n^2$

a = 2, b = 2, f(n) = $n^2$ then $n^{log_b^a} = n^{log_2^2}$ < f(n)

T(n) = Theta( $n^2$ )