

311 HW4

Q1

a) ~~Let~~ Let $P1 = [a, b, \dots, u, v]$

$$P2 = MCP(a, u)$$

$P2 + (u, v)$ should be $P1$

Prove by contradiction

Assume $P2 + (u, v) \neq P1$

then $Capacity(P2 + (u, v)) > Capacity(P1)$

but this contradicts $P1$ is MCP

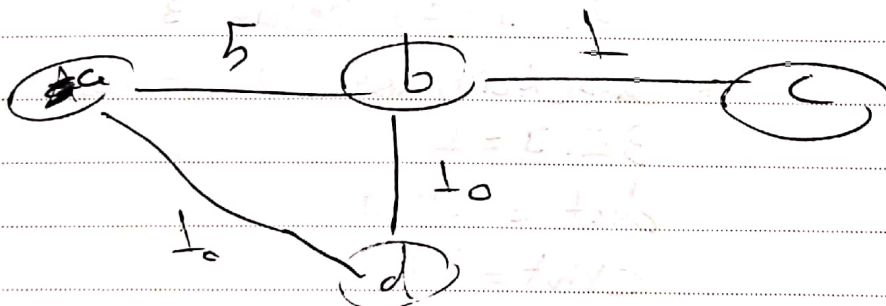
So such $P2$ can't exist

And we prove the question

b)

~~Yes, as we can find another path $P1$ then $MCP(a, v)$ could be $P1$~~

No, here's a counter example



$$MCP(a, c) = [a, b, c]$$

$$\text{but } MCP(a, b) = [c, d, b]$$

c) Yes, we can use Dijkstra with few modifications

Modified Dijkstra ($G(V, E), s$)

Init Max heap PA

for $V \in V$ do

$Key[V] = -\infty$

~~Neighbor~~ $Neighbor[V] = null$

$Key[s] = 0$

for $v \in V$ do

Enduser (PA, v) $\#$ Based on Key

while $PA \neq \emptyset$ do

$V := Dequeue(PA)$

for u such that $(u, v) \in E$ do

if $weight(u, v) + Key[v] > Key[u]$ then

$Neighbor[u] = v$

$Key[u] = weight(u, v) + Key[v]$

Update (PA, u)

return $Neighbor[]$

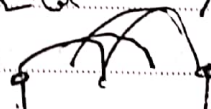
Sketch

Power
Station

into

AC's

Power



Q2

```
def function (A[0...m-1], n, m):
```

```
    def comp(a, b):
```

```
        return a - b ≤ n
```

```
B = boolean array (Size = m)
```

```
cnt = 1 # cnt station
```

```
last = m-1
```

```
dist = 0
```

```
while True
```

```
    for i = cnt → last:
```

```
        if not comp(A[i], dist):
```

```
            Error "Can't cover"
```

```
        if i == last:
```

```
            if A[i] - dist ≤ n/2:
```

```
                return B
```

```
            B[i] = 1; return B
```

```
        if comp(A[i], dist) and not comp(A[i+1], dist):
```

```
            B[i] = 1
```

```
            dist = A[i]
```

```
            cnt = i + 1
```

The algorithm is $\Theta\left(\frac{m}{n}\right)$ where $\frac{m}{n}$ is the number of stations