

PRIM (graph[][], int size , int start) Prim on Adj (الف : سوال ۴)
is $O(V^2)$

$V = \text{array} [\text{size}] ; \Rightarrow$ Create array for keeping Key and π

for $i = 0$ to size $\rightarrow O(|V|) \Rightarrow$ initialize

$V[i].\text{key} = \infty$

$V[i].\pi = \text{nil}$

$V[i].\text{index} = i$

priority - queue $\leftarrow \text{Nil} \Rightarrow$ Creating a priority - queue

\rightarrow based on the $V.\text{key}$

Add ALL V into Priority - queue $\rightarrow O(|V|)$

decrease - key (start, 0) \Rightarrow bring the root to top

while (pq \neq null) $\rightarrow O(|V|)$

$u \leftarrow \text{extract - min}()$

(iterate on \Leftarrow for $i = 0$ to size $\rightarrow O(|V|)$
all other vertices)

\Leftarrow if $V[i] \in \text{pq} \ \&\& \ \text{graph}[u.\text{index}][i] < V[i].\text{key}$

[$V \in Q \ \& \ w(u, V) < \text{key}(V)$] $V[i].\pi = u.\text{index}$

و حالا به سراغ کاهش کلید می رویم

decrease - key ($V[i]$, $\text{graph}[u.\text{index}][i]$)

چون هر رأس حداقل ۱ بار extract می شود پس داریم :

$$T = |V| + |V| + (|V|) (|V| + \log |V|)$$

عملیات کاهش

$$\in O(|V|^2)$$

(I) اوله اینله برای گرفتن MST لازم است که تمامی یال‌ها از لوج به بزرگ مرتب شوند.

(II) تنها حالتی که می‌توان درخت‌های مختلفی گرفت این است که وزن حاد ۲ تا از یال‌ها

بسیار باشد. ← MST

(II) : در صورت برابری وزن ۲ یال، کافی است تا یالی که حتماً خواهیم در MST قرار گیرد را

قبل از یال دیگر با وزن برابر قرار دهیم.

چون KRUSKAL هنگام طی کردن یال‌ها به ترتیب از لوج به بزرگ آن‌ها را بررسی می‌کند

یال انتخابی، در صورتی که در MST حضورش لازم باشد (یعنی راس‌های آن توسط

یال‌های دیگر قبلاً اضافه نشده باشند) اولویت دارد و می‌تواند درخت مورد نظر ما را به

ما ببرد.

(I) : اگر یال‌های ما بر اساس وزن مرتب شده باشند به صورت صعودی؛ آنگاه KRUSKAL

در هر مرحله یالی با کمترین وزن را در صورت لزوم برای Tree انتخاب می‌کند. پس در پایان آن

با یال‌های داده شده حتماً یک MST از گراف اصلی خواهیم داشت.