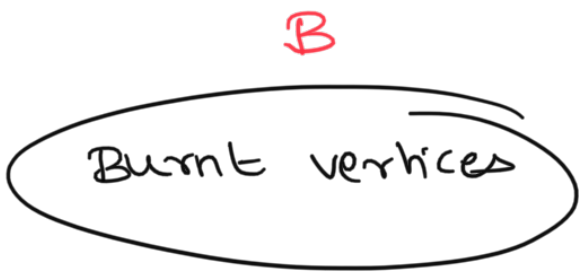
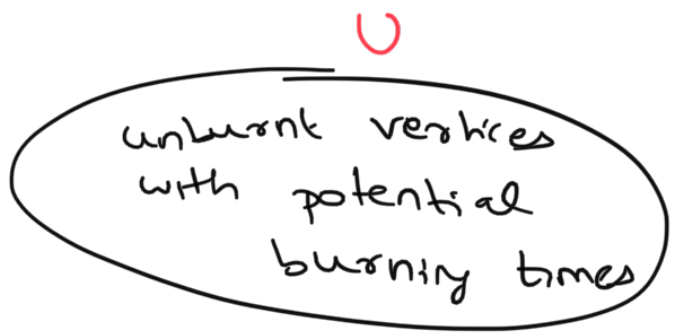
Dijkstra with heaps

Recall Dijkstra algorithm

I





Initially $B = \emptyset$

$U = \{$ start vertex : 0 time $\}$

II Pick vertex in U with min time

Delete it from U,

Add it to B (potential time : actual time it was burnt)

Update potential burning times of newly burnt vertices

III : Repeate step II until $U = \emptyset$

Implementation using heaps

Maintain U as a heap !!

(min heap : parent value $\leq$ its children's values)

Complexity analysis

V : vertex set E : edge set

Have to burn each vertex, $|V|$ of them

→ "Burning" = deleting min of heap

→ $O(\log |\text{size of heap}|)$ operation

$|\text{size of heap}| \leq |V|$ at every step

→ so $\leq O(|V| \log |V|)$

Have to update neighbours' potential burn times of each burnt vertex

→ over all, $O(|E|)$ # $\wedge$ updates.

→ Each update takes $O(\log |\text{size of heap}|)$

so $\leq O(|E| \log |V|)$

so net complexity is $\leq O(|V| \log |V| + |E| \log |V|)$