Lec16 贪婪算法、最小生成树

Graphs (review)

Digraph $G=(V,E)$
- set V of vertices (singular: vertex)
- set $E \subseteq V \times V$ of edges

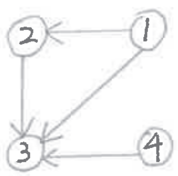
Undirected Graph: E contains unordered pairs

$|E| = O(|V|^2)$

if G is connected (连通的), $|E| > |V|-1$

Graph Representations

- adjacency matrix of $G=(V,E)$, where $V=\{1,2,\cdots,n\}$, is the $n \times n$ matrix A given by $A[i,j] = \begin{cases} 1, & \text{if } (i,j) \in E \\ 0, & \text{if } (i,j) \notin E \end{cases}$



	1	2	3	4
1	0	1	1	0
2	0	0	1	0
3	0	0	0	0
4	0	0	1	0

$\Theta(|V|^2)$ storage $\Longrightarrow$ dense representation

- adjacency list of $v \in V$ is the list $Adj[v]$ of vertices adjacent to v

$Adj[1] = \{2,3\}$
$Adj[2] = \{3\}$
$Adj[3] = \{\ \}$
$Adj[4] = \{3\}$

sparse!

$|Adj[v]| = \begin{cases} \text{degree}(v), & \text{undirected} \\ \text{out-degree}(v), & \text{directed} \end{cases}$

Handshaking Lemma (undirected graph)

$$\sum_{v \in V} \text{degree}(v) = 2|E|$$

结点也占据存储

for undirected graphs $\Longrightarrow$ adjancy list representation uses $\Theta(|V|+|E|)$ storage, it is basically the same thing asymptotically for digraphs

Minimum Spanning Trees

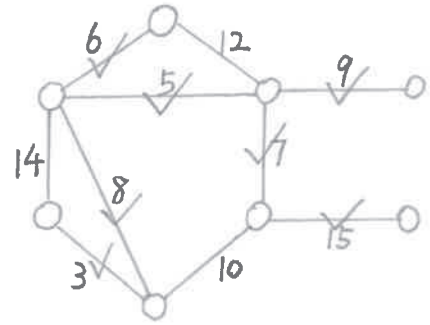
input: connected undirected graph $G=(V,E)$ with weight function $w: E \rightarrow R$

for simplicity, assume all edge weights are distinct

output: a spanning tree T (connects all the vertices) at minimum weight

$$W(T) = \sum_{(u,v)\in T} w(u,v)$$

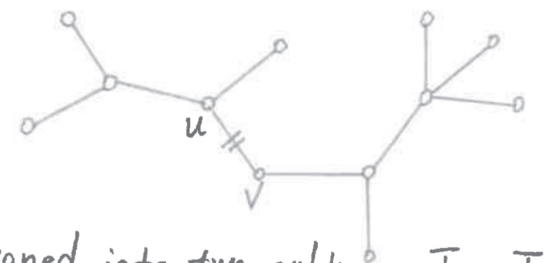
Example:



Optional Substructure

MST T:
(other edges in the graph will not be shown)

remove an arbitrary edge (u,v) in the MST T



then T is partitioned into two subtrees T_1, T_2

Theorem:

T_1 is MST for $G_1 = (V_1, E_1)$,

G_1 is a subgraph of G induced by the vertices in T_1, i.e.

V_1 = vertices in T_1, $E_1 = \{(x,y) \in E,\ x,y \in V_1\}$

similarly for T_2

proof: Cut & Paste

$w(T) = w(u,v) + w(T_1) + w(T_2)$

if T_1' is better than T_1 for G_1,

then $T' = \{(u,v)\} \cup T_1' \cup T_2$ would be better than T for G

conflication!

Overlapping Subproblems?

YES!

Can we use Dynamic Programming?

YES!

but it turns out that MST exhibits a powerful property that, we call, the hallmark for greedy algorithms

Hallmark for Greedy Algorithms (这是可应用某算法的证据)

greedy-choice property:

a locally optimal choice is globally optimal

Theorem:

let T be MST of $G=(V,E)$, and let $A \subseteq V$

suppose $(u,v) \in E$ is the least weight edge connecting A to $V-A$

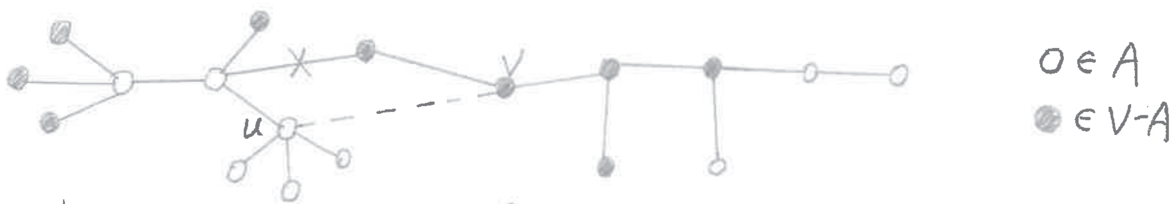
then, $(u,v) \in T$

Proof:

suppose $(u,v) \notin T$

Cut & Paste

T:



consider unique simple path from u to v in T,

swap (u,v) with the first edge on this path that connects a vertex in A to a vertex in $V-A$

图中的'x'边

a lower weight spanning tree than T, resulting a contradiction

Prim's Algorithm

idea: to maintain $V-A$ as a priority queue Q,
to key each vertex in Q with the weight of the least-weight edge
(vt. 赋值)
connecting it to a vertex in A

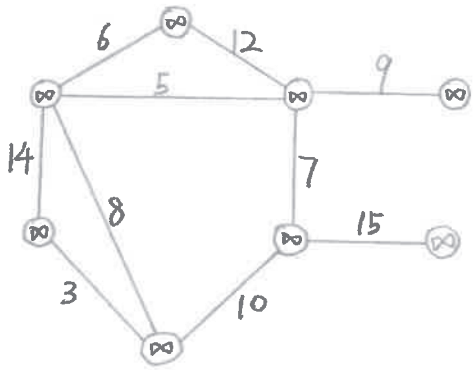
pseudo-code:

$Q \leftarrow V \quad (A \leftarrow \emptyset)$
$v.key \leftarrow \infty, \forall v \in V$
$s.key \leftarrow 0$ for arbitrary $s \in V$

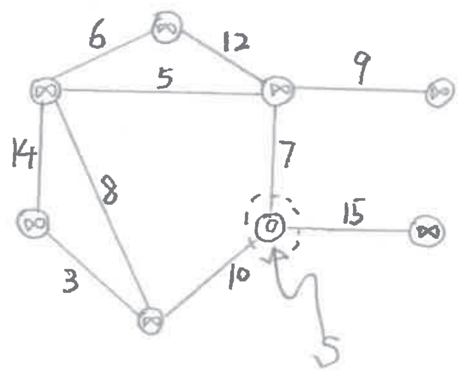
while $Q \neq \emptyset$
  do $u \leftarrow$ Extract-Min(Q)
  for each $v \in Adj[u]$
    do if $v \in Q$ and $w(u,v) < v.key$
      then $v.key \leftarrow w(u,v)$
      $\pi[v] \leftarrow u$ (隐含了一个降低键值的操作)

at the end, $\{(v, \pi[v])\}$ forms MST

Example:

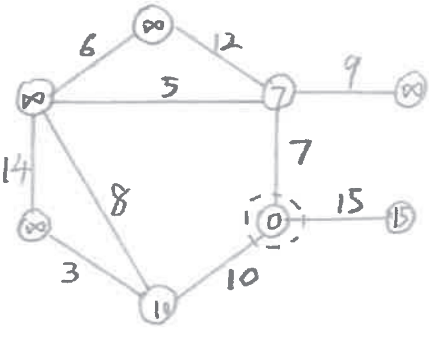


select S →

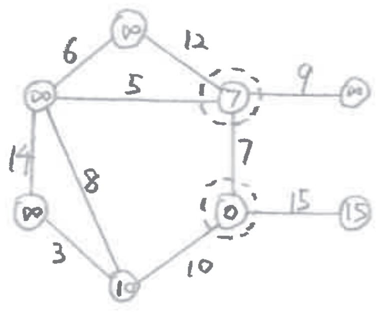


◌:表示已加入A

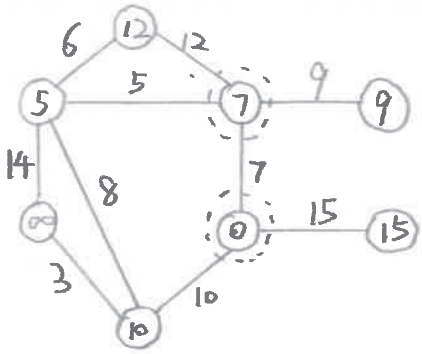
更新距离 →



取出"最便宜"的一个 →



更新距离



→ ...

Analysis

handshaking $\Rightarrow \theta(E)$ implicit decrease keys

$$\text{time} = \theta(|V| \cdot T_{\text{Exact-Min}} + |E| \cdot T_{\text{Decrease-Keys}})$$

$$= \begin{cases} O(|V|^2), & \text{array} \leftarrow \text{dense 用这个} \\ O(|E|\lg|V|). & \text{binaryheap} \leftarrow \text{sparse 用这个} \end{cases}$$