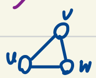
題目： 給定一非負整數 k 和一 complete graph $G=(V,E)$

定義一 cost function 為 $c: V\times V \longrightarrow \mathbb{Z}^+ \cup \{0\}$ 滿足：

triangle inequality：$\forall u,v,w \in V,\ c(u,w) \le c(u,v)+c(v,w)$



問是否存在一個 cost 至多 k 的 traveling salesman tour?

Algorithm： Approx_ETSP(G,c)

1. 求 G 上之一個 MST，令其 root 為 r
2. 由 r 開始以 preorder 順序走訪 T，每次走訪至新的點時，加入該點至 H
   最終再將 r 加入 H
3. return H

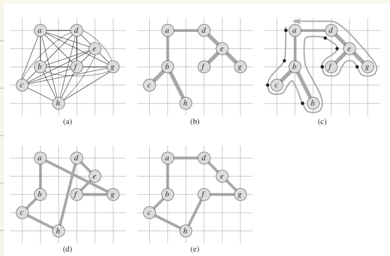
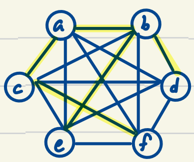
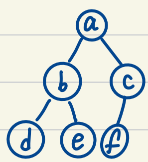


Figure 35.2 The operation of APPROX-TSP-TOUR. **(a)** A complete undirected graph. Vertices lie on intersections of integer grid lines. For example, f is one unit to the right and two units up from h. The cost function between two points is the ordinary euclidean distance. **(b)** A minimum spanning tree T of the complete graph, as computed by MST-PRIM. Vertex a is the root vertex. Only edges in the minimum spanning tree are shown. The vertices happen to be labeled in such a way that they are added to the main tree by MST-PRIM in alphabetical order. **(c)** A walk of T, starting at a. A full walk of the tree visits the vertices in the order $a,b,c,b,h,b,a,d,e,f,e,g,e,d,a$. A preorder walk of T lists a vertex just when it is first encountered, as indicated by the dot next to each vertex, yielding the ordering a,b,c,h,d,e,f,g. **(d)** A tour obtained by visiting the vertices in the order given by the preorder walk, which is the tour H returned by APPROX-TSP-TOUR. Its total cost is approximately 19.074. **(e)** An optimal tour H^* for the original complete graph. Its total cost is approximately 14.715.

Example：

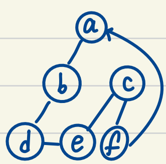


1. 利用 Prim's Algo 找到 MST



2. 利用 DFS，preorder 走訪並加入 H

$\therefore$ H 為：



故：$\langle a\ b\ d\ e\ c\ f\ a \rangle$ 為一 traveling salesman tour!

Note： 若實際以 preorder 走訪的話

得到之 walk 會含重複點為 $W=\langle a\ b\ d\ b\ e\ b\ a\ c\ f\ c\ a \rangle$

可知：H 為 W 之子序列，稱 W 為 full walk

且 $w(H) \le w(W)$ by 三角不等式

Theorem： Approx_ETSP 會為 ETSP 上之 2-approximation algorithm

① Approx-ETSP 之執行時間為 $O(|V|\lg|V|+|E|)=O(|V|^2)$
   故為 poly-time 之 Algorithm

② $\forall E' \subseteq E$，令 $c(E')=\sum_{(u,v)\in E'} c(u,v)$，又令 $T.E$ 為 T 上邊所成集合
   設 H^* 為 G 上之 optimal traveling salesman tour
   而 $c(H^*) \ge c(T.E)$ $(\because G^*$ 去掉一個 edge 為 spanning tree，又 T 為 MST$)$

令 W 為 T 上之 full walk，則 $c(W) = 2c(T.E)$ ($\because$ full walk 會走訪每个 T.E 每个邊各2次)
又已知：$\forall u,v,w \in V,\ c(u,w) \le c(u,v) + c(v,w)$

$\because$ H 為 W 之子序列，所以：$c(H) \le c(W) = 2c(T.E) \le 2c(H^*)$
故 $\frac{c(H)}{c(H^*)} \le 2$ 故 $\rho(n) = 2$

Theorem: 若 $P \neq NP$，則 TSP 之 ρ-approx algo 必不存在，其中 $\rho \ge 1$ 為任意 contant

Proof The proof is by contradiction. Suppose to the contrary that for some number $\rho \ge 1$, there is a polynomial-time approximation algorithm A with approximation ratio ρ. Without loss of generality, we assume that ρ is an integer, by rounding it up if necessary. We shall then show how to use A to solve instances of the hamiltonian-cycle problem (defined in Section 34.2) in polynomial time. $\because HC \le_p TSP$ Since Theorem 34.13 tells us that the hamiltonian-cycle problem is NP-complete, Theorem 34.4 implies that if we can solve it in polynomial time, then P = NP.

Let $G = (V, E)$ be an instance of the hamiltonian-cycle problem. We wish to determine efficiently whether G contains a hamiltonian cycle by making use of the hypothesized approximation algorithm A. We turn G into an instance of the traveling-salesman problem as follows. Let $G' = (V, E')$ be the complete graph on V; that is,

$$E' = \{(u, v) : u, v \in V \text{ and } u \neq v\} .$$

Assign an integer cost to each edge in E' as follows:

$$c(u, v) = \begin{cases} 1 & \text{if } (u, v) \in E , \\ \rho|V| + 1 & \text{otherwise} . \end{cases}$$

We can create representations of G' and c from a representation of G in time polynomial in $|V|$ and $|E|$.

Now, consider the traveling-salesman problem (G', c). If the original graph G has a hamiltonian cycle H, then the cost function c assigns to each edge of H a cost of 1, and so (G', c) contains a tour of cost $|V|$. On the other hand, if G does not contain a hamiltonian cycle, then any tour of G' must use some edge not in E. But any tour that uses an edge not in E has a cost of at least

$$\begin{aligned} (\rho|V| + 1) + (|V| - 1) &= \rho|V| + |V| \\ &> \rho|V| . \end{aligned}$$

Because edges not in G are so costly, there is a gap of at least $\rho|V|$ between the cost of a tour that is a hamiltonian cycle in G (cost $|V|$) and the cost of any other tour (cost at least $\rho|V| + |V|$). Therefore, the cost of a tour that is not a hamiltonian cycle in G is at least a factor of $\rho + 1$ greater than the cost of a tour that is a hamiltonian cycle in G.

Now, suppose that we apply the approximation algorithm A to the traveling-salesman problem (G', c). Because A is guaranteed to return a tour of cost no more than ρ times the cost of an optimal tour, if G contains a hamiltonian cycle, then A must return it. If G has no hamiltonian cycle, then A returns a tour of cost more than $\rho|V|$. Therefore, we can use A to solve the hamiltonian-cycle problem in polynomial time. ■

若 G 上有 HC，則 $v^* = |V|$
又 Approx_Algo 產生之 $v \le \rho v^*$
$\le \rho|V|$
而非 HC 之 $v^* > \rho|V|$
$\therefore$ 若 G 上有 HC，A 必可得到 HC
則 $HC \in P \Rightarrow NP = P$ ($\rightarrow\leftarrow$)

(中央100): <u>Euclidean Traveling Salesman Problem</u>

给定一 Complete graph $G=(V,E)$, $w: V\times V \longrightarrow Z^{+} \cup \{0\}$ 且 $|V|=n$

欲找到一 cycle C^{*} 走訪各 vertex 各一次, 且具最小之 weight, 而 T^{*} 為 G 上 MST

①. show that: $w(T^{*}) \le w(C^{*})$

② 给定 T^{*}, 設計一 2-approximation algorithm 來找 C

---

①. $\because$ C^{*} 去除任意一邊皆為 spanning tree $\therefore$ $w(C^{*}) \ge w(T)$, 其中: T 為 spanning tree

又 T^{*} 為 MST, $w(T^{*}) \le w(T)$, $\forall T$, 故 $w(T^{*}) \le w(C^{*})$

②. <u>idea:</u> 找到一 C 為 $\frac{w(C)}{w(C^{*})} \le 2 \Rightarrow w(C) \le 2w(C^{*})$

<u>Algorithm:</u> ETSP(G, T^{*})

1. 設 r 為 T^{*} 之 root, 由 r 開始以 preorder 順序走訪 T^{*}
2. 走訪至新的點時, 加入該點至 H
3. 走訪完後加入 r 至 H
4. return HC H

令 W 為 T^{*} 上之 full walk, 則 $w(W) = 2w(T^{*})$

$\therefore$ $w(C) \le 2w(T^{*})$

又 $w(T^{*}) \le w(C^{*})$

$\therefore$ $w(C) \le 2w(C^{*}) \Rightarrow \rho(n) = 2$ *.