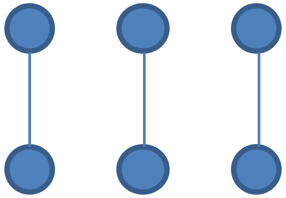
- 作业讲解
  - JH第4章练习4.3.2.3、4.3.2.6、4.3.2.9、4.3.3.5、4.3.3.6

JH第4章练习4.3.2.3

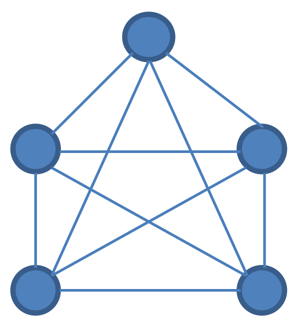
Exercise 4.3.2.3. (a) Find a subgraph G' of the graph G in Fig. 4.3 such that Algorithm 4.3.2.1 can output a vertex cover whose cardinality is 6 while the cardinality of an optimal vertex cover for G' is 3.

(b) Find, for every positive integer n, a Graph G_n with a vertex cover of the cardinality n, but where Algorithm 4.3.2.1 can compute a vertex cover of the size $2n$. □



JH第4章练习4.3.2.6

Exercise 4.3.2.6. Construct an infinite family of graphs for which Algorithm 4.3.2.1 always (independent of the choice of edges from E) computes an optimal vertex cover. □



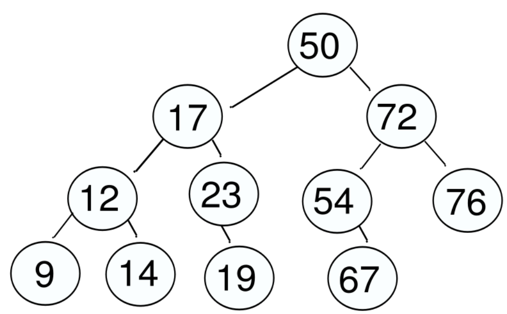
K_{2n+1}

为什么K_{2n}不行？

JH第4章练习4.3.2.9

Exercise 4.3.2.9. (a) Design a polynomial-time algorithm for MIN-VCP when the set of input graphs is restricted to the trees.
(b) Design a polynomial-time d-approximation algorithm for MIN-VCP with $d < 2$ when the input graphs have their degree bounded by 3. □

- 两点观察
  - 一定有一个最小点覆盖不包含任何叶子
    （否则，用其父顶点替换，仍是最小点覆盖）
  - 如果叶子不在点覆盖中，其父顶点必须在
- 算法设计
  - 反复地：任选一叶子的邻点（父顶点），删除关联的边



JH第4章练习4.3.2.9

Exercise 4.3.2.9. (a) Design a polynomial-time algorithm for MIN-VCP when the set of input graphs is restricted to the trees.

(b) Design a polynomial-time d-approximation algorithm for MIN-VCP with $d < 2$ when the input graphs have their degree bounded by 3. □

- 算法设计
  - 反复地：选择覆盖最多剩余边的顶点
- 近似比证明
  - 参考Lemma 4.3.2.12

- 教材讨论
  - JH第4章第3节第5小节

问题1：算法4.3.5.1

- 算法的基本思路
- 算法近似比证明的基本思路
- 相对误差最坏的例子

Algorithm 4.3.5.1.

Input: A complete graph $G = (V, E)$, and a cost function $c : E \to \mathbb{N}^+$ satisfying the triangle inequality

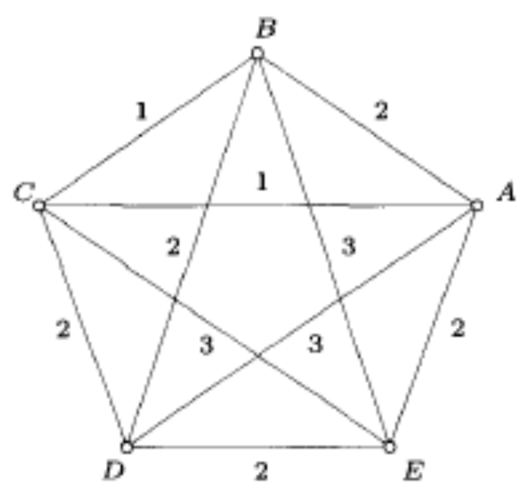
$$c(\{u, v\}) \leq c(\{u, w\}) + c(\{w, v\})$$

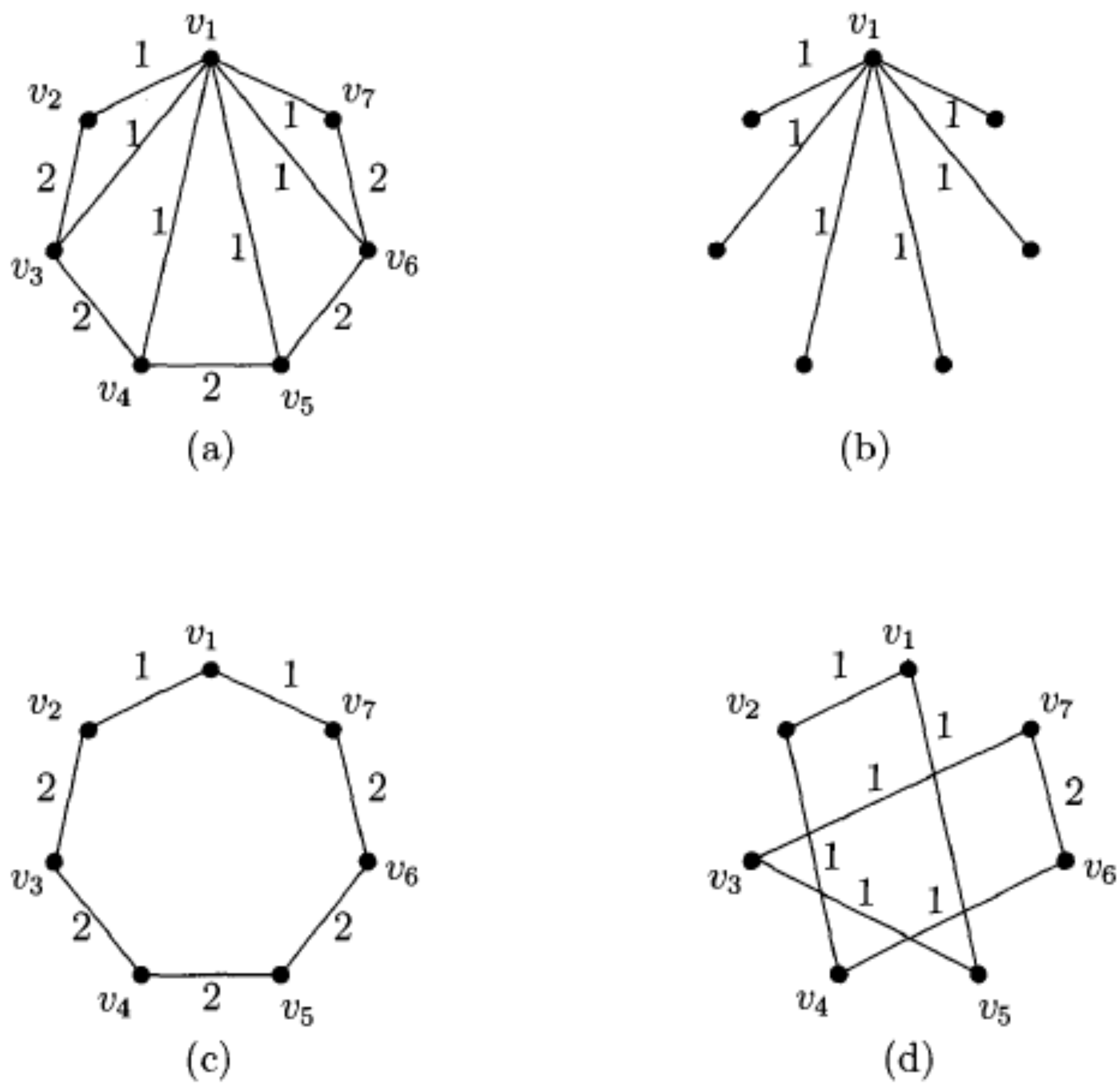
for all three different $u, v, w \in V$ {i.e., $(G, c) \in L_\triangle$}.

Step 1: Construct a minimal spanning tree T of G according to c.

Step 2: Choose an arbitrary vertex $v \in V$. Perform depth-first-search of T from v, and order the vertices in the order that they are visited. Let H be the resulting sequence.

Output: The Hamiltonian tour $\overline{H} = H, v$.





v_1
v_2
v_3
v_4
v_5
v_6
v_7
1
2
(a)
(b)
(c)
(d)

问题2：算法4.3.5.4

- 算法的基本思路
- 算法近似比证明的基本思路
- 相对误差最坏的例子

Algorithm 4.3.5.4. CHRISTOFIDES ALGORITHM

Input: A complete graph $G = (V, E)$, and a cost function $c : E \to \mathbb{N}^+$ satisfying the triangle inequality.
Step 1: Construct a minimal spanning tree T of G according to c.
Step 2: $S := \{v \in V \mid deg_T(v) \text{ is odd}\}$.
Step 3: Compute a minimum-weight[21] perfect[22] matching M on S in G.
Step 4: Create the multigraph $G' = (V, E(T) \cup M)$ and construct an Eulerian tour ω in G'.
Step 5: Construct a Hamiltonian tour H of G by shortening ω (i.e., by removing all repetitions of the occurrences of every vertex in ω in one run via ω from the left to the right).
Output: H.

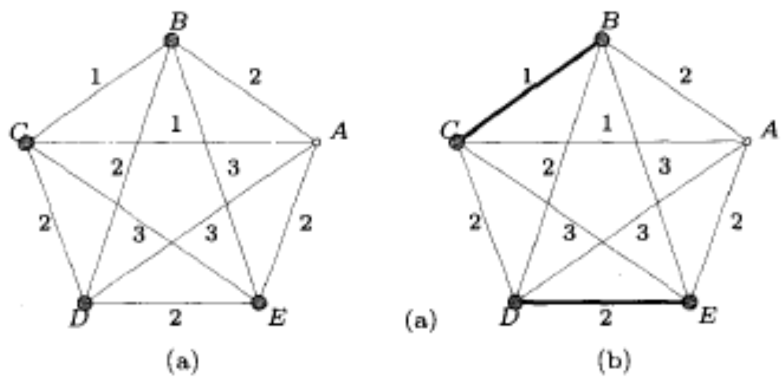


Fig. 4.10.

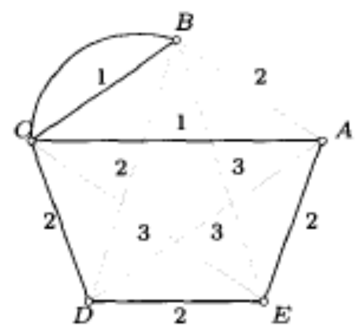
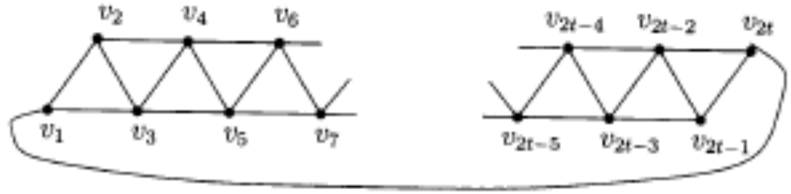
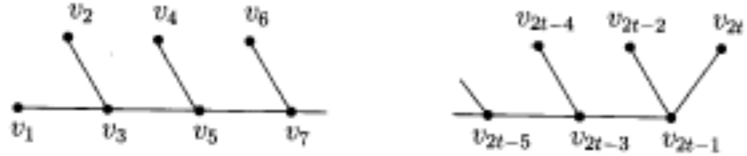


Fig. 4.11.


v_2 v_4 v_6
v_{2t-4} v_{2t-2} v_{2t}
v_1 v_3 v_5 v_7
v_{2t-5} v_{2t-3} v_{2t-1}

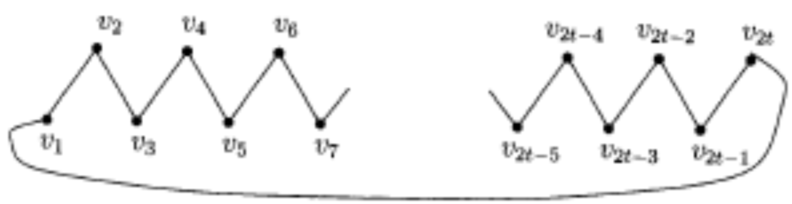
(a)


v_2 v_4 v_6
v_{2t-4} v_{2t-2} v_{2t}
v_1 v_3 v_5 v_7
v_{2t-5} v_{2t-3} v_{2t-1}

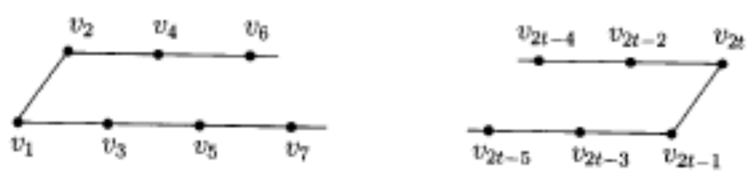
(b)


v_2 v_4 v_6
v_{2t-4} v_{2t-2} v_{2t}
v_1 v_3 v_5 v_7
v_{2t-5} v_{2t-3} v_{2t-1}

(c)


v_2 v_4 v_6
v_{2t-4} v_{2t-2} v_{2t}
v_1 v_3 v_5 v_7
v_{2t-5} v_{2t-3} v_{2t-1}

(d)


v_2 v_4 v_6
v_{2t-4} v_{2t-2} v_{2t}
v_1 v_3 v_5 v_7
v_{2t-5} v_{2t-3} v_{2t-1}

(e)

- Δ-TSP的不可近似性
  - Papadimitriou and Vempala (2006): 220/219
  - Lampis (2014): 185/184
  - Karpinski, Lampis, and Schmied (2015): 123/122

  (gap)

  - Christofides (1976): 3/2

问题3：算法4.3.5.18

- 算法的基本思路
- 算法近似比证明的基本思路
- 算法的意义（尽管近似比并不很好）

Algorithm 4.3.5.18. SEKANINA'S ALGORITHM

Input: A complete graph $G = (V, E)$, and a cost function $c : E \to \mathbb{N}^+$.
Step 1: Construct a minimal spanning tree T of G according to c.
Step 2: Construct T^3.
Step 3: Find a Hamiltonian tour H in T^3 such that $P_T(H)$ contains every edge of T exactly twice.
Output: H.

Theorem 4.3.5.19. SEKANINA'S ALGORITHM *is a polynomial-time* 2*-approximation algorithm for* $\triangle$-TSP.

Proof. Obviously, Step 1 and 2 of SEKANINA'S ALGORITHM can be performed in time $O(n^2)$. Using Lemma 4.3.5.17 one can implement Step 3 in time $O(n)$. Thus, the time complexity of SEKANINA'S ALGORITHM is in $O(n^2)$.

Let H_{Opt} be an optimal solution for an input instance (G, c) of $\triangle$-TSP. Following the inequality (4.32) we have $cost(T) \leq cost(H_{Opt})$. The output H of SEKANINA'S ALGORITHM can be viewed as shortening the path $P_T(H)$ by removing repetitions of vertices in $P_T(H)$. Since $P_T(H)$ contains every edge of T exactly twice,

$$cost(P_T(H)) = 2 \cdot cost(T) \underset{(4.32)}{\leq} 2 \cdot cost(H_{Opt}). \tag{4.51}$$

Since H is obtained from $P_T(H)$ by exchanging simple subpaths by an edge, and c satisfies the triangle inequality,

$$cost(H) \leq cost(P_T(H)). \tag{4.52}$$

Combining (4.51) and (4.52) we obtain $cost(H) \leq 2 \cdot cost(H_{Opt})$. □

问题4：TSP问题实例的划分

- 如何对TSP问题的所有实例进行划分？
  - dist
  - *p*-strengthen triangle inequality