算法设计与分析

随机算法. 155103163 宋博宇.

$\boxed{9.1}$ (1). 总能求得一个正确的解，所以是舍伍德算法.

(2). 随机在S中选择一个元素s

$P(s > \min(s,k)) = \frac{n-k}{n}$. 考察集合长度: $k, k+1 \cdots\cdots n-1$.

$P(s < \min(s,k)) = \frac{k-1}{n}$ 考察集合长度: $n-(k-1), n-(k-2) \cdots\cdots n-1$

$P(s = \min(s,k)) = \frac{1}{n}$. 直接输出s.

$$E(|S|) = \frac{n-k}{n} \cdot \frac{k+(k+1)+\cdots+(n-1)}{n-k} + \frac{k-1}{n} \cdot \frac{n-(k-1)+n-(k-2)+\cdots\cdots n-1}{k-1}$$

$$= \frac{(n-k)(n-1+k)}{2n} + \frac{(2n-k)(k-1)}{2n}$$

$$= \frac{n^2+(2k-3)n-2k^2+2k}{2n} = \frac{n}{2} + \frac{2k-3}{2} + \frac{k-k^2}{n} \cancel{\le \frac{n}{2}}. \ ①$$

~~所以存在常数b<1，使得算法递归过程集合大小期望为bn，其中b=1/2~~.

$E(k) = \frac{n+1}{2}$ $E(k^2) = \frac{(n+1)(2n+1)}{6}$ 代入上式得

$$E(|S|) = \frac{n}{2} + \frac{n-2}{2} - \frac{n^2-1}{3n}$$

$$= \frac{2n}{3} + \frac{1}{3n} - 1 < \frac{2}{3}n.$$

所以存在常数b<1，使得考察集合大小期望为bn，其中$b=\frac{2}{3}$.

(3) 定义随机变量 $X_k = I\{|S_1| = k-1\}$ $E(X_k)=\frac{1}{n}$

$$T(n) = \sum_{k=1}^{n} X_k \left(T(\max\{k-1, n-k\}) + O(n)\right)$$

$$E(T(n)) = \sum_{k=1}^{n} \frac{1}{n} \cdot E\left(T(\max\{k-1, n-k\})\right) + O(n).$$

$$\le \frac{2}{n} \sum_{k=\frac{n}{2}}^{n-1} E(T(k)) + O(n).$$

$$E(T(n)) = O(n).$$

需要考察的平均长度为 $\frac{n-1+k}{2}$. $\frac{2n-k}{2}$. 0.

$$T(n) = \frac{n-k}{n} T\left(\frac{n-1+k}{2}\right) + \frac{k-1}{n} \cdot T\left(\frac{2n-k}{2}\right) + \theta(n).$$

当k=1或n时为算法最坏情况，考察集合长度最大. 用master定理 a<

$\cancel{T(n)=T(n/2)}$ k=1或n时 $T(n) = \frac{n-1}{n} \cdot T(\frac{n}{2}) + \theta(n)$.

$f(n) = \Omega(n^{\log_b a + \varepsilon})$ $\varepsilon=1$. $T(n)=\theta(n)$.

对于 $k=1,2,\cdots n$. $T(n)=O(n)$.

9.2 假设 $m+n=L$.

```
Random(P.Q.R). // P[i]表示指数为i的系数.
  for i=1 to n.        // n表示测试次数.
      k=Rand(1.~~~~).
      sum=0.
      for j=0 to k.
          sum = sum + P[j]*Q[k-j].
      if sum != R[K]
          return False.
  return True.
```

当 L 很大的时候 即 $n<L$ 时.

若 sum != R[K]. 算法返回 False. 正确率为 100%.

若 返回 True. 则有 $\frac{n}{L}$ 的概率得到正确答案.

该算法是 Monte carlo 算法.

9.3 随机产生 r 个数. 生成一个 $r\times 1$ 列向量 X. 分别计算 $A(B\cdot X)$ 和 $C\cdot X$. 验证是否相等. ~~如果相 执行n遍~~. 若都相等则认为 $A\cdot B=C$. 否则认为 $A\cdot B\neq C$.

时间复杂度: $T(A\cdot(B\cdot X)) = \max\{pq.\ qr\}$.

$T(C\cdot X) = pr$.

所以 $T = O(\max\{pr.pq.qr\})$.

```
Random(A.B.C).
    生成 r 个数.存储为 X[1:r].
    for i=1 to q.
        for j=1 to r
            Y[i] += B[i.j] * X[j].

    for i=1 to p
        for j=1 to r.
            Z[i] += C[i.j] * X[j]

    for i=1 to p.
        for j=1 to q.
            if (A[i.j] * Y[j] != Z[j]).
                return False.
return True.
```

该算法是 Monte carlo 算法.

① 当返回 False 时. 正确率为 100%.

② 当返回 True 时 有可能 $A\cdot B\neq$

正确率为 $\frac{q}{p+r}$.

```
for m=0 to n   // n为验证
    随机选择 i<q j<r
        if A[i]行 * B[j]列 ≠
            return False.
return True.
```

正确率为 $\frac{n}{p\times r}$.

9.4 引理1：若输出的是最小割，则生成树中有且只有一条边是最小割中边。

① 若生成树中没有最小割中边，则生成树不是连通的

② 若生成树中有两条及以上最小割边，则删除一条边后，生成树中还有一条在cut中

引理2：若输出的是最小割，则删除的那条边一定是在最小割中。

若被删除的边不在cut中，这条边将点集分成两部分，且在原图中应为最小割，矛

定理：若满足1、2，则输出的cut为最小割。

$$P(\text{输出最小割}) = P(\text{引理1}) \times P(\text{引理2}) = \frac{k-1}{\frac{kn}{2}} \times \frac{1}{n-1} = \frac{2k-2}{k} \cdot \frac{1}{n(n-1)} = \Omega($$

（k = 最小割边数，n = 点个数）

9.6 N = 冒泡排序算法需交换的倒置元素个数

N_i = 第 i 个元素需交换的倒置元素个数.

$$E(N) = E\left(\sum_{i=1}^{n} N_i\right) = \sum_{i=1}^{n} E(N_i) = \sum_{i=1}^{n} \frac{1}{n} \sum_{j=0}^{i} j.$$

$$= \frac{1}{2}\left[n^2 + n - n(n+1) + \frac{(n+1)(2n+1)}{6} - \frac{n+1}{2}\right] = \frac{n^2-1}{6}$$

9.7 随机产生一个数 $x = \mathrm{Rand}(0, z)$. $y = z - x$.

计算 $F(z \bmod n) = F(x) + F(y) \bmod m$.

$F(x)$ 和 $F(y)$ 正确可能性为 $\frac{4}{5}$. 因此 $F(z)$ 正确率为 $\left(\frac{4}{5}\right)^2 = \frac{16}{25}$.

运行3次：① $F(x)$ × $F(y)$ ×. $\frac{1}{25}$

② $F(x)$ $F(y)$ 有一个对 $\frac{8}{25}$.

③ $F(x)$ ✓ $F(y)$ ✓. $\frac{16}{25}$.

两次以上得到正确结果：正确率 $= C_3^2 \left(\frac{16}{25}\right)^2 \frac{9}{25} + C_3^3 \left(\frac{16}{25}\right)^3 \approx 0.7$

近似算法

9.1

利用匈牙利算法,可以知道当图为树而非森林时,由于匈牙利算法也是求最小点覆盖,不过 其获得边的个数是最小点个数,因此此时近似比为 2,当图为森林时,近似比会小于 2。综 上可知,利用匈牙利算法求得的最小点覆盖是近似比为 2 的算法。

9.2

首先确定,每台机器上至少要有 2 个以上的任务才可以。设机器的数量为 m 台,任务数量 为 2n+1,每两个任务的执行时间为 m+1,m+2...,2m-1 并且有三个任务的执行时间为 m 此时贪心算法的时间为 4m-1,而最优解的代价为 3m,因此近似比为 4/3-1/3m,可知近似比 为 4/3

9.3

A 表示一个 tsp 问题的最优解,因为(V,A)是连接的,包含了生成树 T,因此 w(A)>=w(T),

B 表示最优解的完全图的边集合,根据三角不等式,我们知道 w(A)>=w(B),由完全匹配可知 w(A)/2>=w(B)/2,设 e1,e2...为欧拉路,此时 w(M)+ w(T)≤w(A)+ w(A)/ 2 和三角不等式算法 3/2- approximative

9.4

设$(X = R \cup S, d)$为 steiner 树问题,并且 T=(V,E)是一棵树,$R \subseteq V \subseteq X$.就会有一个 T′是 R 的生成树 $cost_d(T_0) \leq 2 * cost_d(T)$。

9.5

假设S_k的元素就是集合X的前$d = |S_k|$个元素,即$S_k = \{x_1, \ldots x_d\}$,并且可以假设这些元素是按第 一次选取到包含它们的集合的顺序排列的。这样做只是对X里的元素重新命名以简化描述，因此不失一般性.解得Cx的和是小于等于$H(|S_k|)*w(k)$。最后解出带权集合覆盖的近似比为 ln(n+1)

9.6

令 C*是图 G 的最小 k=|T|路割的最优解。显然 C*是满足如下条件的 k 个割的并集:令 G’=(V,E,C*),则 G’中有 k 个连通分量,每个连通分量中包含恰好包含 T 中的一个点。

$$\begin{aligned} w(E') &= \sum_{i=1}^{k} w(C_i) \\ &\leq \sum_{i=1}^{k} w(C_i^*) \\ &= 2w(C^*) \\ \Rightarrow \frac{w(E')}{w(C^*)} &\leq 2 \end{aligned}$$

即得出一个近似比为 2 的多路割算法。

9.8

整数规划问题可以表示成如下:

Max z=$\sum_{v(V} w(v)|x(v)$

s.t. x(u) + x(v) >=1 for all v

x(v) [0,1] for all v

任意的(u,v)属于E,由约束条件知x(u)+x(v)>=1,因此x(u)和x(v)至少有一个大 于等于1/2,即u,v 或两者在C 中,C是G的一个顶点覆盖。可知近似比为2.

9.10

(1). 下面线性规划的目标函数最优取值 z_{LP} 是 z^* 的上界

$$\begin{aligned} \max \quad & z = \sum_{x_i \in V} x_i \\ \text{s.t.} \quad & x_i + x_j \geq w_{ij} \quad \forall (i,j) \in E \\ & x_i \geq 0 \quad \forall i \in V \end{aligned}$$

证明 假设 (LP) 的最优解为 $x_1^*, \ldots, x_{|V|}^*$，最大匹配问题的最优解为 $e_1^*, \ldots, e_k^*$ 其 $k = |M^*|$ 为最优匹配中的边的个数，且 $e_i^* = (u_i, v_i)$，那么有

$$\begin{aligned} z_{LP} &= x_1^* + x_2^* + \cdots + x_{|V|}^* \\ &= (x_{u_1} + x_{v_1}) + (x_{u_2} + x_{v_2}) + \cdots + (x_{u_k} + x_{v_k}) + \cdots (\text{未被覆盖的顶点}) \\ &\geq (x_{u_1} + x_{v_1}) + (x_{u_2} + x_{v_2}) + \cdots + (x_{u_k} + x_{v_k}) \\ &= w(e_1^*) + w(e_2^*) + \cdots + w(e_k^*) \\ &= z^* \end{aligned}$$

(2). 利用贪心算法得到的 M 构造上述线性规划的可行解 x，证明该可行解的代价至多为 2
据此证明 $2z \geq z^*$
证明

(a) 构造可行解
对于 $(i,j) \in E$ 是被贪心算法选中的边，令 $x_i = x_j = w_{ij}$；
对于其余的 x_j，令 $x_j = 0$

(b) 证明上述解的可行性

如果它被贪心算法选中，有 $x_i + x_j \geq w_{ij}$ 显然成立；
否则，贪心算法处理 $(i,j) \in E$ 时，该边中有一个顶点已经被某条选中的边 $(i,j$
(i',j) 所覆盖。根据贪心算法的处理顺序，知 $w_{ij} \leq w_{i'j}|w_{ij'}$，故 $x_i + x_j = w_{i'j}|w_{ij'} \geq$
成立。

(c) 代价计算

$$
\begin{aligned}
& x_1 + x_2 + \cdots + x_{|V|} \\
= \ & 2 \cdot \textstyle\sum_{(i,j)\text{被贪心算法选中}} w_{ij} \\
= \ & 2z
\end{aligned}
$$

由以上知 $2z \geq z^*$，即近似比为 2 成立

9.11

近似比为 2-1/m,因为 m>2,因此最短平行调度不存在 3/2-近似算法