节选第一、二章

1.1. The following data yield the arrival times and service times that each customer will require,for the first 13 customer will require, for the first 13 customers at a single server system. Upon arrival, a customer either enters service if the server is free or joins the waiting line. When the server completes work on a customer, the next one in line enters service.

Arrival Times: 12 31 63 95 99 154 198 221 304 346 411 455 537

Service Time: 40 32 55 48 18 50 47 18 28 54 40 72 12

(a) Determine the departure times of these 13 customers

(b) Repeat(a) when there are two serves and a customer can be served by either one

(c) Repeat(a) under the new assumption that when the server completes a service, the next customer to enter service in the one who has been waiting the least time

翻译: 以下给出了 13 个客户需要的到达时间和服务时间, 到达后, 客户要么在服务器空闲时进入服务, 要么加入等待队列。当服务器完成对客户的工作时, 下一个排队的客户进入服务

到达时间: 12 31 63 95 99 154 198 221 304 346 411 455 537

服务时间: 40 32 55 48 18 50 47 18 28 54 40 72 12

(a) 确定这 13 位客户的离开时间

(b) 当有两个服务并且客户可以由其中一个服务时

(c) 重复 (a), 使得当服务器完成服务时, 下一个进入服务的客户是等待时间最少的客户

Answers:

(a) 题解析:

第 n 位	服务是否存在空闲	离开时间
1	$0 < 12$, 是	$0 + 12 + 40 = 52$
2	$52 > 31$, 否	$52 + 0 + 32 = 84$
3	$84 > 63$, 否	$84 + 0 + 55 = 139$
4	$139 > 95$, 否	$139 + 0 + 48 = 187$
5	$187 > 99$, 否	$187 + 0 + 18 = 205$
6	$205 > 154$, 否	$205 + 0 + 50 = 255$
7	$255 > 198$, 否	$255 + 0 + 47 = 302$
8	$302 > 221$, 否	$302 + 0 + 18 = 320$
9	$320 > 304$, 否	$320 + 0 + 28 = 348$
10	$348 > 346$, 否	$348 + 0 + 54 = 402$
11	$402 < 411$, 是	$402 + (411 - 402) + 40 = 451$
12	$451 < 455$, 是	$451 + (455 - 451) + 72 = 527$
13	$527 < 537$, 是	$527 + (537 - 527) + 12 = 549$

(b) 题解析:

第 n 位	服务器 A 是否空闲	服务器 B 是否空闲	离开时间
1	$0 < 12$, 是	—	$0 + 12 + 40 = 52$
2	$52 > 31$, 否	$0 < 31$, 是	$31 + 0 + 32 = 63$
3	$52 < 63$, 是	—	$63 + 0 + 55 = 118$
4	$118 > 95$, 否	$63 < 95$, 是	$95 + 0 + 48 = 143$
5	$118 > 99$, 否	$143 > 99$, 否	$99 + (118 - 99) + 18 = 136$
6	$136 < 154$, 是	—	$154 + 0 + 50 = 204$
7	$204 > 198$, 否	$143 < 198$, 是	$198 + 0 + 47 = 245$
8	$204 < 221$, 是	—	$221 + 0 + 18 = 239$
9	$239 < 304$, 是	—	$304 + 0 + 28 = 332$
10	$332 < 346$, 是	—	$346 + 0 + 54 = 400$
11	$400 < 411$, 是	—	$411 + 0 + 40 = 451$
12	$451 < 455$, 是	—	$455 + 0 + 72 = 527$
13	$527 < 537$, 是	—	$537 + 0 + 12 = 549$

(c) 题解析: 代码见 P3 图 1, 运行截图见 P4 图 2

```
#include<bits/stdc++.h>
```

```cpp
using namespace std;

const int N = 13;
deque<int> arrive;
deque<int> service;
deque<int> wait;

int main() {
    int tmp = 0;
    int n = N, m = N;
    while(n--) {
        cin >> tmp;
        arrive.push_back(tmp);
    }
    while(m--) {
        cin >> tmp;
        service.push_back(tmp);
    }
    arrive.push_back(1e9);
    service.push_back(1e9);
    int finishTime = arrive[0];
    int i = 0;
    while(i < N){
        while(finishTime >= arrive[i]){
            wait.push_back(service[i]);
            i++;
        }
        while(finishTime < arrive[i]) {
            if(!wait.empty()) {
                finishTime += wait[wait.size() - 1];
                wait.pop_back();
            } else {
                finishTime += service[i];
```

```
                }
                cout << "离开时间是" << finishTime << endl;
                if(finishTime < arrive[i] && wait.size() == 0) {
                    finishTime = arrive[i];
                }
            }
        }
    }
```

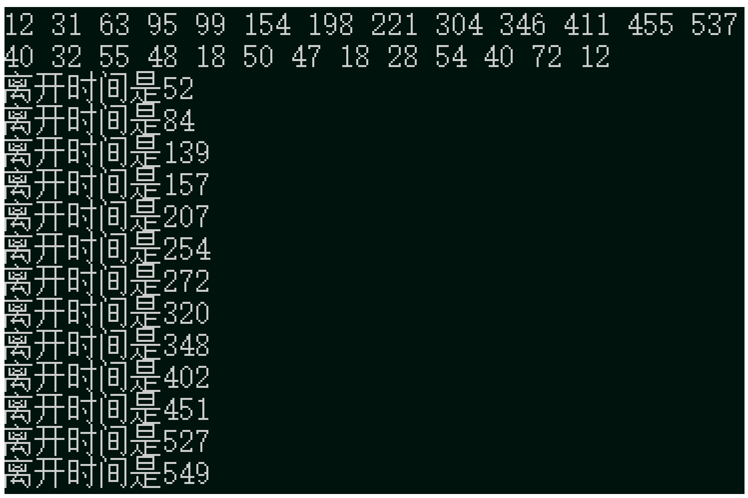

图 1: 第一题 (c) 运行截图

1.2. Consider a service station where customers arrive and are served in their orders of arrival. Let A_n, S_n and D_n denote, respectively, the arrival time, the service time, and the departure time of customer n. Suppose there is a single server and the system is initially empty of customers.

(a) With $D_0 = 0$, argue that for $n > 0$

$$D_n - S_n = Maximum\{A_n, D_{n-1}\}$$

(b) Determine the corresponding recursion formula when there are two servers.

(c) Determine the corresponding recursion formula when there are k servers.

(d) Write a computer program to determine the departure times as a function of the arrival and service times and use it to check your answers in parts (a) and (b) of Exercise 1.

翻译: 一个服务站按照顾客的到达顺序提供服务。设 A_n、S_n 和 D_n 分别表示客户 n 的到达时间、服务时间和离开时间。假设只有一台服务器, 系统最初没有客户

(a) 若 $D_0 = 0$, 讨论当 $n > 0$ 时

$$D_n - S_n = Maximum\,\{A_n, D_{n-1}\}$$

(b) 当有两台服务器时，确定相应的递归公式

(c) 当有 k 台服务器时, 确定相应的递归公式

(d) 编写一个计算机程序, 根据到达和服务时间推算顾客的离开时间, 并用它检查练习 1(a) 和 (b) 部分的答案.

Answer:

(a) 题解析:

$D_n - S_n$ 的实际含义是服务器开始给顾客提供服务的时间, 而这又取决于两点: 顾客何时来以及来了是否需要等侯, 因此完全取决于 D_{n-1}(上一个顾客离开的时间) 和 A_n(顾客到达时间) 哪个大 ($D_{n-1} > A_n$ 说明顾客需要等, 服务开始的时间就等于 D_{n-1}; $D_{n-1} \leq A_n$ 说明顾客不需要等, 服务开始的时间就等于 A_n)

(b) 题解析:

答案: 本题用文字解释很简单,$D_n - S_n$(即顾客开始办理业务的时间), 取决于两点: 一个是 A_n(即顾客到达的时间), 人没到肯定办不了业务; 另一个是取决于两台服务器正在服务的客户当中最先办理好业务的那位所离开的时间, 最后再与 A_n 取个最大值, 就是用户开始办理业务的时间。需要注意的是: 假如客户 k 需要等待, 意味着两台服务器均有人, 但是正在办理业务的两个人不一定是 $k-1$ 和 $k-2$, 可能发生以下这种情况: 就是 A 和 B 两台服务器, 其中的 A 服务器正在为顾客 $k-3$ 服务, 但是这人很麻烦, 需要办理十小时的业务, 但是后面进来的 $k-2$ 和 $k-1$ 顾客, 办理业务都只需要一分钟, 等 $k-2$ 顾客在 B 服务器办理完业务的时候,$k-1$ 只能去 B 服务器, 因为 A 服务器还在为 $k-3$ 客户服务. 甚至, 等客户 $k-1$ 办理完业务的时候, 顾客 $k-3$ 依旧未完成业务办理, 因此: 顾客 k 必须等第 $k-1$ 位顾客在服务器 B 完成业务办理后, 才能去 B 服务器接受业务办理. 因此,$D_n - S_n$ 并不是取决于 $Maximum\,\{D_{n-1}, D_{n-2}\}$, 而是取决于两台服务器正在服务的客户当中最先

办理好业务的那位所离开的时间, 正在接受服务的两个人, 不一定是 $k-1$ 和 $k-2$, 也不一定是 $k-3$ 和 $k-1$, 有很多种可能, 甚至可能出现以下这种极端情况: 第一个顾客进来后, 一直办理服务直到当天服务器下班才结束, 这样, 对于其他后面进来办理业务的顾客来说, 服务站就退化为了只有一台服务器, 因此必须等到第 $k-1$ 位顾客离开第 k 位才能开始办理业务. 但可以确认的是: 第 k 位顾客想要开始办理业务的前置条件是: 第 $k-1$ 位顾客已经在接受业务办理了, 意思就是第 k 位顾客不可能比第 $k-1$ 位顾客更先开始办理业务.

因此递推公式为:

$$D_n - S_n = Max\left\{A_n, min(D_{n-1}, D_{n-r})\right\}\left((r \in N^+)\bigcap(r \in [2,3,4...n-1])\right)$$

(c) 题解析:

思路和 (b) 一模一样, 不再赘述, 有 k 台服务器意味着第 n 个顾客开始办理业务的时间取决于正在办理业务的 k 个顾客最先离开的那位, 当然还取决于第 k 个顾客自己到达服务站的时间, 最后两者再取个最大值, 就是 $D_n - S_n$ 的值.

因此递推公式为:

$$D_n - S_n = Max\left\{A_n, min(D_{n-1}, D_{n-r_1}, D_{n-r_2}, ..., D_{n-r_{k-1}})\right\}$$

且需要满足:

$$((r_1, r_2, ..., r_{k-1} \in N^+)\bigcap(r_1, r_2, ..., r_{k-1} \in [2,3,4...n-1])$$
$$((\forall i, j \in (N^+\bigcap[2,3,4...n-1]))\bigcap(i \neq j)), r_i \neq r_j$$

(d) 题解析:

```
#include<bits/stdc++.h>
using namespace std;

const int M = 100005;
int K, N;
deque<int> arrive;
deque<int> service;
```

```
deque<int> wait;
deque<int> isService;
int finishTime[M];

int main() {
    int tmp = 0;
    cin >> K >> N;
    int n = N, m = N;
    while(n--) {
        cin >> tmp;
        arrive.push_back(tmp);
    }
    while(m--) {
        cin >> tmp;
        service.push_back(tmp);
    }
    int use = 0;
    int NowTime = 0;
    for(int i = 0; i < N; i++) {
        NowTime = arrive[i];
        sort(isService.begin(), isService.end());
        while(NowTime > isService[0] && isService.size()) {
            isService.pop_front();
            use--;
        }
        while(wait.size() && use < K) {
            isService.push_back(wait[0]);
            wait.pop_front();
            use++;
        }
        sort(isService.begin(), isService.end());
        int f = 0;
        if(use < K) {
```

```
            f = arrive[i] + service[i];
            isService.push_back(f);
            use++;
        } else {
            if(wait.size() >= K) {
                f = service[i] + wait[wait.size() - K];
            } else {
                f = service[i] + isService[wait.size()];
            }
            wait.push_back(f);
        }
        finishTime[i] = f;
    }
    for(int v = 1; v <= N; v++) {
        cout << ”第” << v << ”位顾客的离开时间是”
        cout << finishTime[v-1] << endl;
    }
}
```

第一题 (a) 运行截图

```
1 13
12 31 63 95 99 154 198 221 304 346 411 455 537
40 32 55 48 18 50 47 18 28 54 40 72 12
第1位顾客的离开时间是52
第2位顾客的离开时间是84
第3位顾客的离开时间是139
第4位顾客的离开时间是187
第5位顾客的离开时间是205
第6位顾客的离开时间是255
第7位顾客的离开时间是302
第8位顾客的离开时间是320
第9位顾客的离开时间是348
第10位顾客的离开时间是402
第11位顾客的离开时间是451
第12位顾客的离开时间是527
第13位顾客的离开时间是549
```

图 2: 第一题 (a)C++ 代码

第一题 (b) 运行截图

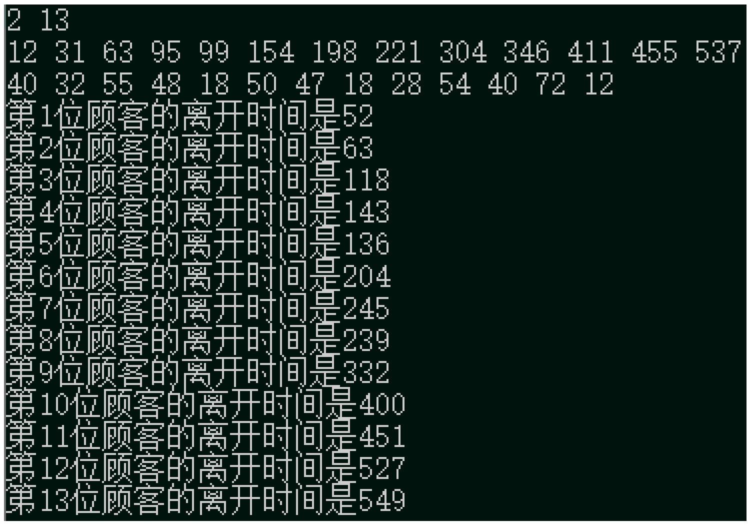

图 3: 第一题 (b)C++ 代码

2.29. Persons A,B,and C are waiting at a bank having two tellers when it opens in the morning.Persons A and B each go to a teller and C waits in line. If the time it takes to serve a customer is an exponential random variables with parameter λ, what is the probability that C is the last to leave the bank?
翻译: 早上银行开门时,A、B 和 C 三个人正在一家有两个出纳员的银行等候。A 和 B 各自去出纳员那里,C 排队等候。如果为客户服务所需的时间是一个参数为 λ 的指数随机变量, 那么 C 最后离开银行的概率是多少?
Answer:

$$\begin{aligned} P(C_{last}) &= 2P(A_{beforeB}\&\&B_{beforeC}) \\ &= 2P(B_{beforeC}|A_{beforeB})P(A_{beforeB}) \end{aligned}$$

由于指数分布的无记忆性,A 在 B 之前离开后,B 和 C 谁先走谁后走的概率都是 $\frac{1}{2}$, 因此:$P(B_{beforeC}|A_{beforeB}) = \frac{1}{2}$, 所以,$P(C_{last}) = 2 * \frac{1}{2} * \frac{1}{2}$

2.32. For a Poisson process with rate λ,find $P\{N(s) = k|N(t) = n\}$ when $s < t$.
翻译: 对于速率为 λ 的泊松过程, 当 $s < t$ 时, 求出 $P\{N(s) = k|N(t) = n\}$.

Answer:

$$\begin{aligned}
P\{N(s)=k|N(t)=n\} &= \frac{P(N(s)=k,N(t)=n)}{P(N(t)=n)}\\
&= \frac{P(N(s)=k,N(t)-N(s)=n-k)}{P(N(t)=n)}\\
&= \frac{P(N(s)=k)P(N(t-s)=n-k)}{P(N(t)=n)}\\
&= \frac{\frac{(\lambda s)^k}{k!}e^{-\lambda s}\frac{(\lambda(t-s))^{n-k}}{(n-k)!}e^{-\lambda(t-s)}}{\frac{(\lambda t)^n}{n!}e^{-\lambda t}}\\
&= \frac{s^k(t-s)^{n-k}}{t^n}\frac{n!}{k!(n-k)!}
\end{aligned}$$

2.33. Repeat Exercise 32 for $s>t$.
翻译: 当 $s>t$ 时,32 题的结果如何?
Answer:

$$\begin{aligned}
P\{N(s)=k|N(t)=n\} &= \frac{P(N(s)=k,N(t)=n)}{P(N(t)=n)}\\
&= \frac{P(N(s)-N(t)=k-n,N(t)=n)}{P(N(t)=n)}\\
&= P(N(s-t))=k-n\\
&= \frac{(\lambda(s-t))^{k-n}}{(k-n)!}e^{-\lambda(s-t)}
\end{aligned}$$

2.36. If X and Y are independent and identically distributed exponential random variables, show that the conditional distributions of X, give that $X+Y=t$, is the uniform distribution on $(0,t)$.
翻译: 如果 X 和 Y 是独立且同分布的指数随机变量, 在给定 $X+Y=t$ 的前提下, 证明随机变量 X 服从 $(0,t)$ 上的均匀分布.
Answer:
由于 X 和 Y 均服从 Exp(λ), 且指数分布同时也是 $\alpha=1$ 的伽马分布, 由伽马分布的可加性, 可知 X+Y 服从 Ga(2,λ)。所以对于 $x\in[0,t]$, 有:

$$P\{X=x|X+Y=t\}=\frac{P(X=x,Y=t-x)}{P(X+Y=t)}=\frac{\lambda e^{-\lambda x}\lambda e^{-\lambda(t-x)}}{\lambda e^{-\lambda t}\frac{\lambda t}{1!}}=\frac{1}{t}$$

因此在给定 $X+Y=t$ 的前提下, 随机变量 X 服从 $(0,t)$ 上的均匀分布