

講義録レポート

講義録コード

04-23-1-305-01

講 座	基本情報技術者	科目①	模試編
目標年	2022年上期合格目標	科目②	模試解説 午後 (問 5・問 6・Python)
コース	本科生 (プラス) 午後実践コース 午後マスター 午後リビジョンコース	回 数	2 回
用 途	ビデオブース ・ テープレクチャー ・ 集合ビデオ WEB ・ DVD通信 ・ 資料通信		

講師名	遠藤 金作 先生 藤澤 大輔 先生	講義録枚数	10 枚	※レポート 含まず サイズ (B5)
		補助レジュメ枚数	0 枚	
		その他	0 枚	

講義構成	解説1 (24分) → 解説2 (56分) → 休憩 (10分) → 解説3 (61分)
使用教材	
配付教材	
備考	※模試解説 午後について、午後①では問 1～問 4、午後②では問 5・問 6・Pythonの解説を行います。
	※Webで実施された方の問題・解答解説につきましては、模試実施後に表示される「結果画面」にてご確認ください。

この講義録の著作権は、TAC株式会社または権利者に帰属しており、当社に無断で複製、改変、転載、転用、インターネット上にアップロードする等の著作権を侵害する行為は法律によって禁止されております。

T A C 情報処理講座

情報処理 講義録	コース・講義等	基本情報	科目	模試解説 午後	回数	2(Python)
----------	---------	------	----	---------	----	-----------

配布物	★テスト類 : [] ★その他の配布物1 : [] ★その他の配布物2 : []	講師	遠藤・藤澤 先生
-----	---	----	----------

黒板内容

問5 マネジメント

設問1 a-ウ 設問2 b-オ, c-ク, d-エ

設問3 e-ウ, f-エ 設問4 g-ウ

設問1

- ・金曜日は"CPU使用率"が高い
- ・月曜日は"ディスク使用率"が高い

↳ 表1より 月曜日は"検索"の比率が高く、
金曜日は"データ登録"の比率が高い

設問2

□b 金曜日のメモリ : $38/60 \times 100 = 63.33... \div 63.3\%$
小数第2位で四捨五入

□c 月曜日のディスク : $46/65 \times 100 = 70.76... \div 70.8\%$

□d 金曜日のCPU : 72.9% Max

月曜日のメモリ : 51.7%

金曜日のディスク : 32.3%

情報処理 講義録

コース・講義等

基本情報

科目

模試解説 午後

回数

2(Python)

配布物

★テスト類： []
★その他の配布物1： []
★その他の配布物2： []

講師

遠藤・藤澤

先生

黒板内容

設問3

㊦

物理的制約

表1の上の記述より

単位時間内の受付トランザクション数に上限を設け、
しきい値を設定する

㊦

価格制約

㊦1の下記述より

ピーク時の時間帯の利用単価を、
他の時間帯よりも高く設定する

設問4

データ登録

0.36 → 0.42 → 0.39 → 0.41

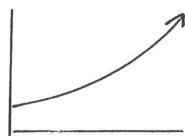
毎年
約0.4ずつ増加



検索

0.19 → 0.24 → 0.29 → 0.38

年々増加幅が大きくなっている



→ 今後さらに
増加幅が大きくなるかも!

㊦

情報処理 講義録

コース・講義等

基本情報

科目

模試解説 午後

回数

2(Python)

配布物

★テスト類 : []
★その他の配布物1 : []
★その他の配布物2 : []

講師

遠藤・藤澤

先生

黒板内容

問6 アルゴリズム

設問1 a-7, b-1, c-7, d-1

設問2 e-7, f-1, g-1, h-1 設問3 i-1

○ fn_summary (ten[], f[], M, N)

○ 整数型: freq_accum[N+1] ... freq_accum[0] ~ [N]

① 全て0で初期化 得点0~Nに対応

② $i: 0, i < M, 1$ (得点ごとの) 度数の集計

・ $\text{freq_accum}[\text{ten}[i]] \leftarrow \text{freq_accum}[\text{ten}[i]] + 1$

ten[i]を添字とした freq_accum[ten[i]] に人数をカウント

図2より

ten[0] = 4 → freq_accum[4] にカウント

ten[1] = 6 → " [6] "

⋮ ⋮ ⋮

ten[99] = 6 → freq_accum[6] にカウント

ten[98] = 9 → " [9] "

ten[99] = 7 → " [7] "

③ $r: 1, r \leq N, 1$ (その得点までの) 累積度数の計算

・ $\text{freq_accum}[r] \leftarrow \text{freq_accum}[r] + \text{freq_accum}[r-1]$

図2より freq_accum[1] = 10 ← 10 + 0

" [2] = 24 ← 14 + 10 - 1点までの人数

" [3] = 30 ← 6 + 24

その得点の人数
+
1つ前の得点までの
累積人数

情報処理 講義録

基本情報

科目

模試解説 午後

回数

2(Python)

配布物

- ★テスト類: []
- ★その他の配布物1: []
- ★その他の配布物2: []

講師

遠藤・藤澤

先生

黒板内容

- $fn_calculate(freq_accum[], f[], M)$
- 整数型: r, idx, mid
- 実数型: pre

(4行) $\cdot r \leftarrow 0$

[C] $idx = 0$ のとき

$\boxed{\text{freq_accum}[t] = 0}$
 $\cdot r \leftarrow r + 1$
 $\cdot f[0] \leftarrow r$

得点の最小値を見つける

$freq_accum$
 $r \downarrow$

0	0
1	10
2	24
...	...

0点はいない
1点か10人
0より大きい値を見つけなければ
 $freq_accum[t] > 0$

[d] $idx = 1 \sim 3$ のとき

$\cdot mid \leftarrow (1 + M) \times idx \div 4$

$\boxed{freq_accum[r] < mid}$ (終了) $freq_accum[t] \geq mid$

$\cdot r \leftarrow t + 1$

mid 番目以上になったら STOP

→ このとき r が mid 番目の得点

$(1 + M) \times idx \% 4 = 0$ (余りが0ならば)

$\cdot f[idx] \leftarrow r$

$\cdot pre \leftarrow r$ 回避

$\boxed{freq_accum[t] < mid + 1}$ $mid + 1$ 番目の得点を t に!

$\cdot r \leftarrow t + 1$

$\cdot f[idx] \leftarrow (pre + r) \div 2$ (mid 番目と $mid + 1$ 番目の得点の平均値)

情報処理 講義録

コース・講義等

基本情報

科目

模試解説 午後

回数

2(Python)

配布物

- ★テスト類: []
- ★その他の配布物1: []
- ★その他の配布物2: []

講師

遠藤・藤澤

先生

黒板内容

設問2 トレース

(1) idx = 0 のとき

r	0	1	...
	2	6	...

freq - accum[0] ≠ 0
①の条件 1回
(r=0 のとき)

(2) idx = 1 のとき

$$(1 + 100) \times 1 \div 4 = 25 \text{ 余り } 1$$

0	1	2	3	4
2	6	11	19	30

25以上の値を探す

③の条件 5回判定 [E]

mid + 1 = 25 + 1 = 26 なので、同じ 30 を見つけて STOP

④の条件 1回 (r=4)

(3) idx = 2 のとき

$$(1 + 100) \times 2 \div 4 = 50 \text{ 余り } 2$$

4	5
30	57

50以上の値を!

③の条件 2回判定 [F]

mid + 1 = 50 + 1 = 51 なので、同じ 57 を見つけて STOP

④の条件 1回 (r=5)

(4) idx = 3 のとき

$$(1 + 100) \times 3 \div 4 = 75 \text{ 余り } 3$$

5	6	7
57	75	84

75以上の値を!

③の条件 2回判定

mid + 1 = 75 + 1 = 76 なので、76以上の値を探す

④の条件 2回判定 [G] (r=7)

情報処理 講義録

コース・講義等

基本情報

科目

模試解説 午後

回数

2(Python)

配布物

★テスト類 : []
★その他の配布物1 : []
★その他の配布物2 : []

講師

遠藤・藤澤

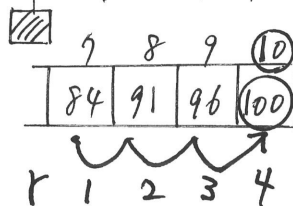
先生

黒板内容

(5) $idx = 4$ のとき

■ $freq_accum[t] < 100$

・ $t \leftarrow t+1$



終了
 $freq_accum[t] \geq 100$
100以上でSTOP

②の条件 4回判定 [h]

⑨ 注) なお、解答は重複して選んでよい ← 必ず
どこかに
重複がある!!

設問3

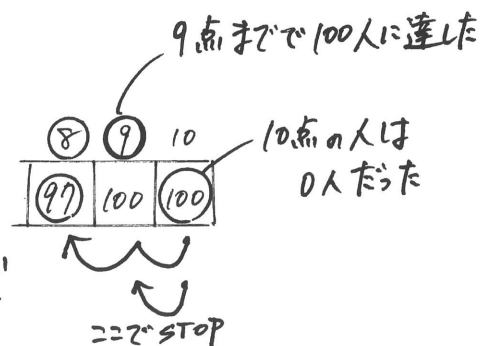
・ $t \leftarrow 10$ (図2より $N=10$ を使用)

■ $freq_accum[t-1] = M$

・ $r \leftarrow t-1$

■

・ $q[4] \leftarrow r$



× $freq_accum[t] \neq M (=100)$ だと1個ずれてしまう
 $freq_accum[t-1] \neq M$ で STOP

情報処理 講義録	コース・講義等	基本情報	科目	模試解説 午後	回数	2(Python)
----------	---------	------	----	---------	----	-----------

配布物	★テスト類 : [] ★その他の配布物1 : [] ★その他の配布物2 : []	講師	遠藤・藤澤 先生
-----	---	----	----------

黒板内容

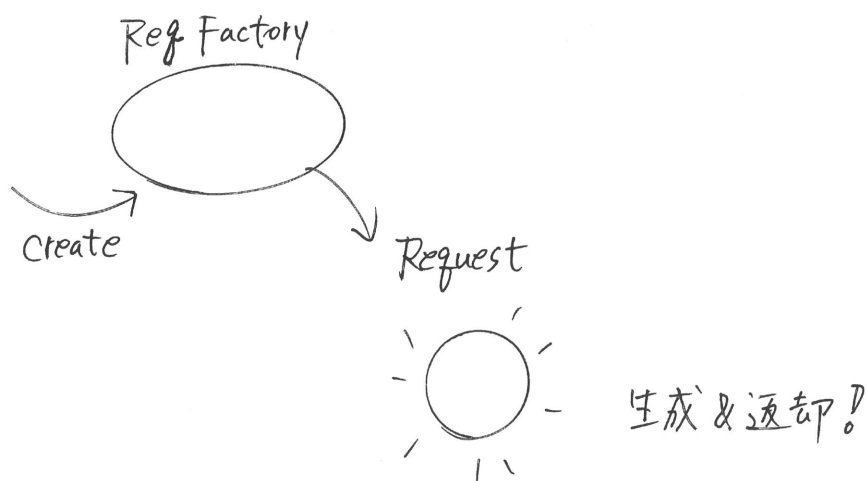
問9 (Python)

待ち行列モデル



1秒に λ_a 件到着

1秒で λ_s 件処理



情報処理 講義録	コース・講義等	基本情報	科目	模試解説 午後	回数	2(Python)
----------	---------	------	----	---------	----	-----------

配布物	★テスト類 : []	講師	遠藤・藤澤 先生
	★その他の配布物1 : []		
	★その他の配布物2 : []		

黒 板 内 容

sim 時刻

stime 0

chl
0
1
 窓口

g
...
 ← 待ち行列 窓口と

fin
...
 ← 完了

rf
create
 → Reg Factory

↓ Request

rq
rq
 →

while

if rq.arrived(~):

{ rq が 着いた

for i in range(~):

if chl[i] != None:

if ~ done

完了の処理

if ~

割り当て

```

def expinterval (~):
    while True:
        val = ~
        yield val a 力
    
```

Request 到着時刻 間隔 時間

~ -- init -- (self, stime, arvint, srv):

self. arr = stime + arvint

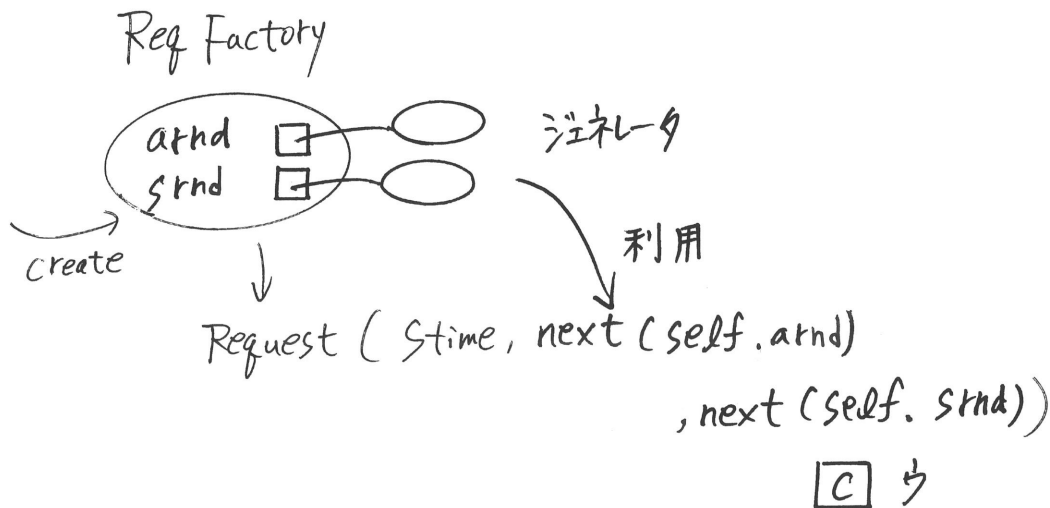
self. strt = -1 開始時刻 set Start Time で設定

get WaitTime では self. strt - self. arr を返す! b エ

情報処理 講義録	コース・講義等	基本情報	科目	模試解説 午後	回数	2(Python)
----------	---------	------	----	---------	----	-----------

配布物	★ テ ス ト 類 : []	講師	遠藤・藤澤 先生
	★ その他の配布物 1 : []		
	★ その他の配布物 2 : []		

黒 板 内 容



while ~

if rq. arrived (~) : ← rq が着いた

rq. setQlen (len(q)) □ ウ

f. append (rq) ← 行列へ □ □ □

窓口で rq = rf. create (~) ← 次の要求

for ~

完了

if chl[i] == None and len(q) > 0

chl[i] = f. pop(0) □ オ

↑ キューの先頭を割り当て!

情報処理 講義録	コース・講義等	基本情報	科目	模試解説 午後	回数	2(Python)
----------	---------	------	----	---------	----	-----------

配布物	★テスト類： []	講師	遠藤・藤澤 先生
	★その他の配布物1： []		
	★その他の配布物2： []		

黒板内容

`maxql = max(fin, key = f).getalen() + 1`

↙ 大小比較の基準

`lambda val: val.getalen()` ①

設問2

48行 `fin.append(((chl[i], i)))`

56, 59行 `total += g.get ~`

`print( ~ { h }, ~ )`

60行の直前で!

`e[0]` ⑦ `e[1]` ⑧

↘ `fin = [e[0] for e ~]`

i ⑧

