# 株式会社フィックスポイント様
# プログラミング試験

# 目次

# 1.言語、実行方法

・言語 c++

。実行方法

・ターミナルから実行をすることを想定

・c++17以降のバージョンでコンパイルするようにオプションの指定をお願いします

(例: g++ -std=c++17 ファイル名)

・入力は標準入力を想定

・始めにログファイルの総数を入力するようにお願いします。

# 2.データの入力について

・入力例

1行目　パラメータ
q　(N)　(m)　(t)

2行目　ログデータ(q行)
<確認日時>,<サーバアドレス>,<応答結果>
<確認日時>,<サーバアドレス>,<応答結果>
<確認日時>,<サーバアドレス>,<応答結果>
.
.
.
<確認日時>,<サーバアドレス>,<応答結果>

入力はすべて整数を想定

q : 入力するログファイルの総数

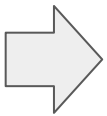
N : タイムアウトの猶予回数
(設問2、設問4で使用)

m : 平均応答時間を計算するサンプル数
(設問3)

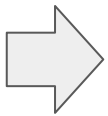
t : サーバが過負荷状態になっていると
判断する閾値 (設問3)

・入力実装方法

ログ1個(1行)を文字列
として受け取る



カンマで文字列を分割



日時、アドレス、結果の
配列にそれぞれ格納

# 3.設問1

## 設問概要

・故障状態のサーバアドレスとそのサーバの故障期間を出力するプログラムを作成せよ。

※故障状態は応答結果がタイムアウトしているもの

## 考えたこと

・ipアドレスをキーとする連想配列で情報管理すればよさそう

・故障期間は復旧(正常応答)したときに出力すればよさそう

・連想配列では故障開始時間とそのサーバが故障していたかのフラグが必要そう

# 3.設問1

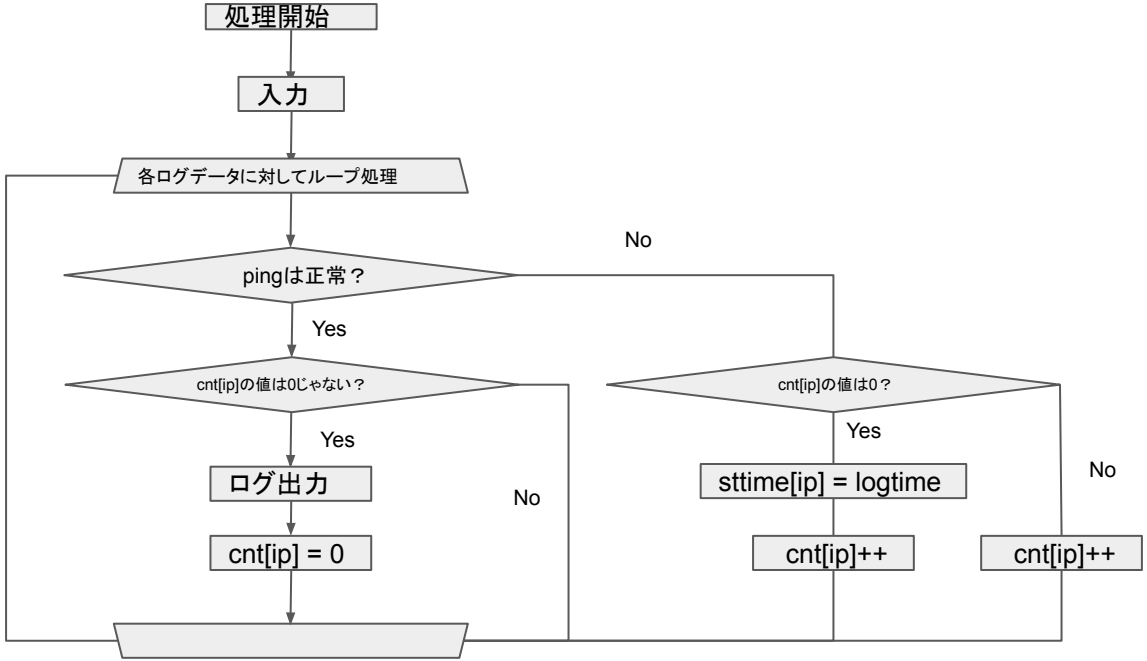
## 変数説明

ipアドレス(文字列)をキーとする
二つの連想配列を用意しました

・cnt : 故障の継続回数を記録
・sttime : 故障の開始時間を記録

他の変数
logtime : 確認日時
ip : サーバのipアドレス
ping : 応答結果

## 簡単な処理手順



# 4.設問2

## 設問概要

・設問1の故障判定に猶予(N回)を持たせよう

## 考えたこと

・設問1のプログラムで既に継続回数を記録してるからログ出力時の処理だけ変えれば

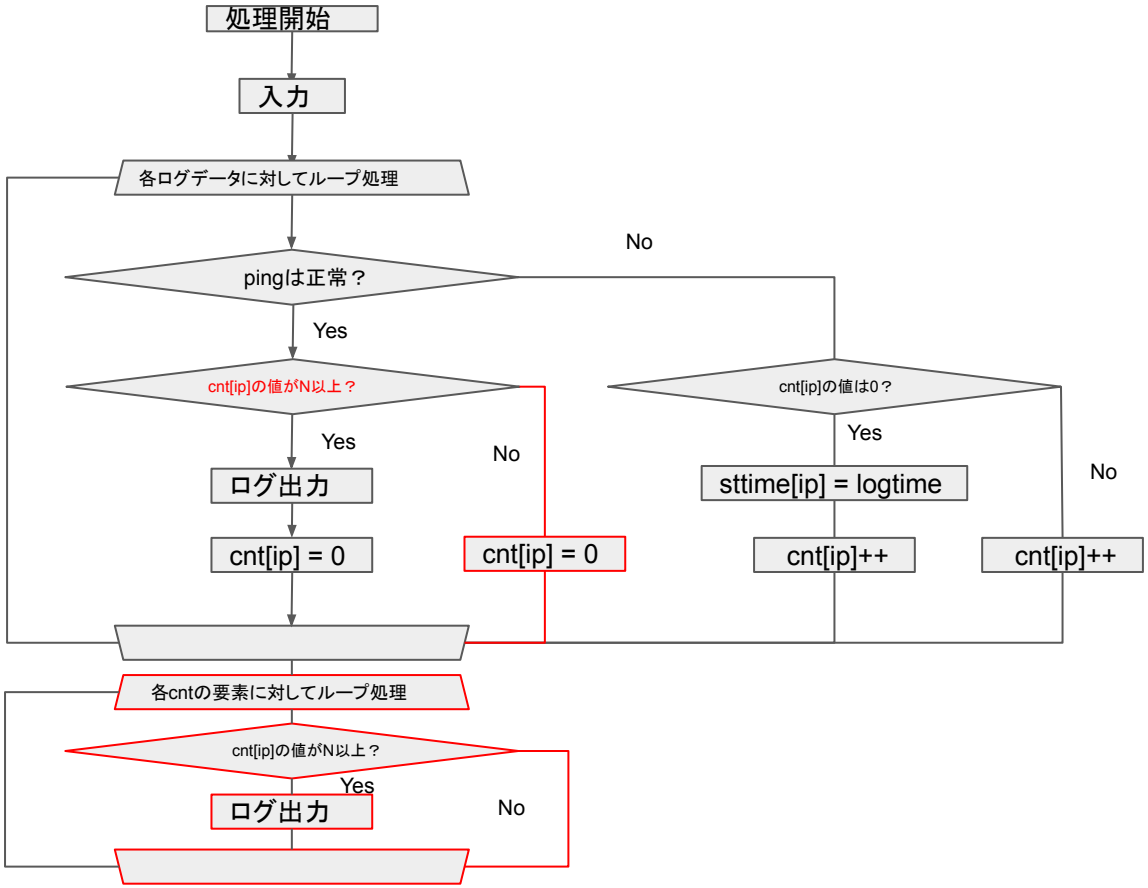
良さそう

・故障検知のニュアンスなら復旧してないのも出力した方がいいかも？

# 4.設問2

・設問1からの変更点を赤で示す

## 簡単な処理手順



# 5.設問3

## 設問概要

・サーバーが過負荷状態か判定したい (過負荷状態かは直近m回の平均応答時間で判断 )

## 考えたこと

・直近m回の応答記録はqueueを使えば良さそう

・m回分の応答時間の総和を保持しておけば O(1)で過負荷状態かの判断ができそう

## 考慮できなかったこと

・タイムアウトしたケースの処理

->平均応答時間を計算する際にどう処理したらいいかわからず

今回はスルーすることにしました (題意に沿うことができず申し訳ございません )

# 5.設問3

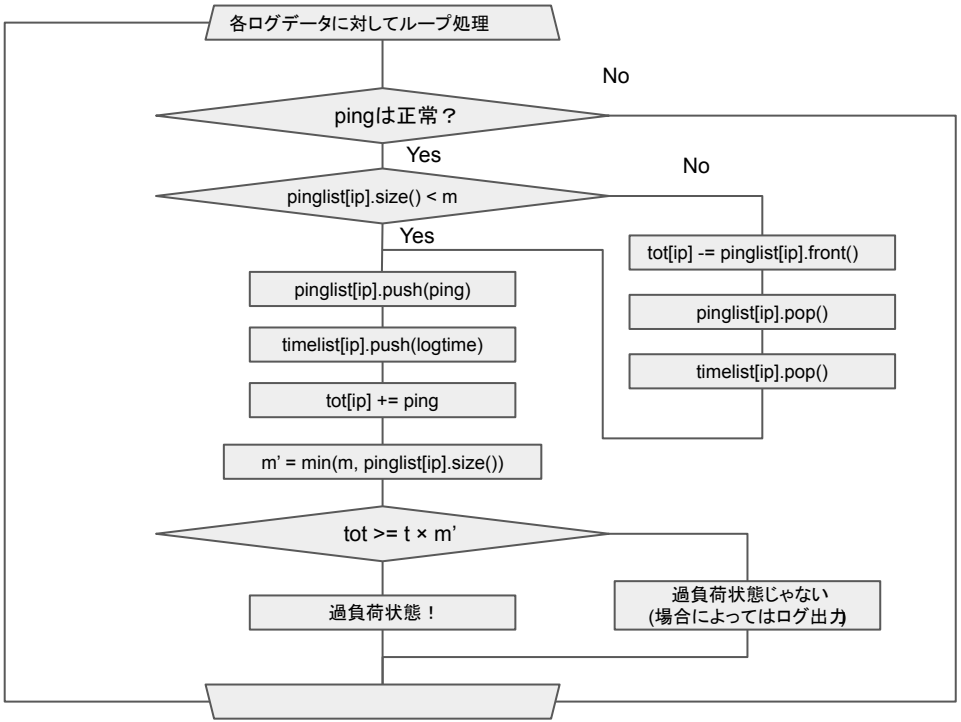
## 変数説明

設問1,2で用いていた
ipアドレス(文字列)をキーとする
2つの連想配列に加えて更に3つの
連想配列を用意

・pinglist : 直近m回の応答時間を記録
(Valueの型 : queue<int>)
・timelist : 直近m回の確認日時を記録
(いらなかったかも)
・tot : 直近m回の応答時間の総和を記録

他の変数
m : 平均応答時間を求める際のパラメータ
(直近何回分使うか)
t : 過負荷状態とみなす際の閾値

## 簡単な処理手順(過負荷状態判定部分)



# 6.設問4

## 設問概要

・サーバー単位ではなくサブネット単位で故障の管理　しよう

## 考えたこと

・連想配列のキーを変えればよさそう

(ipアドレス　->　サブネットのアドレス )

## 考慮できなかったこと

・故障の決め方

作成したプログラム　-> ログを見てあるサブネット内でタイムアウトが連続　N回起こった時

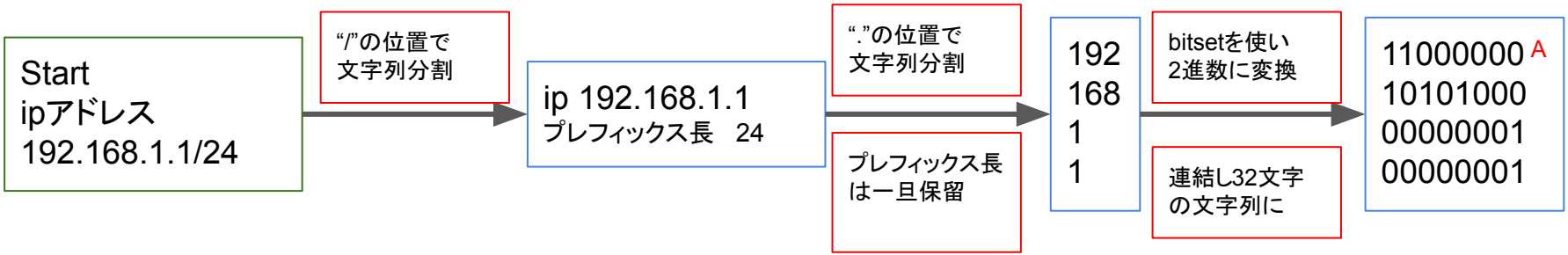
問題文 -> あるサブネット内のサーバが全て故障している時

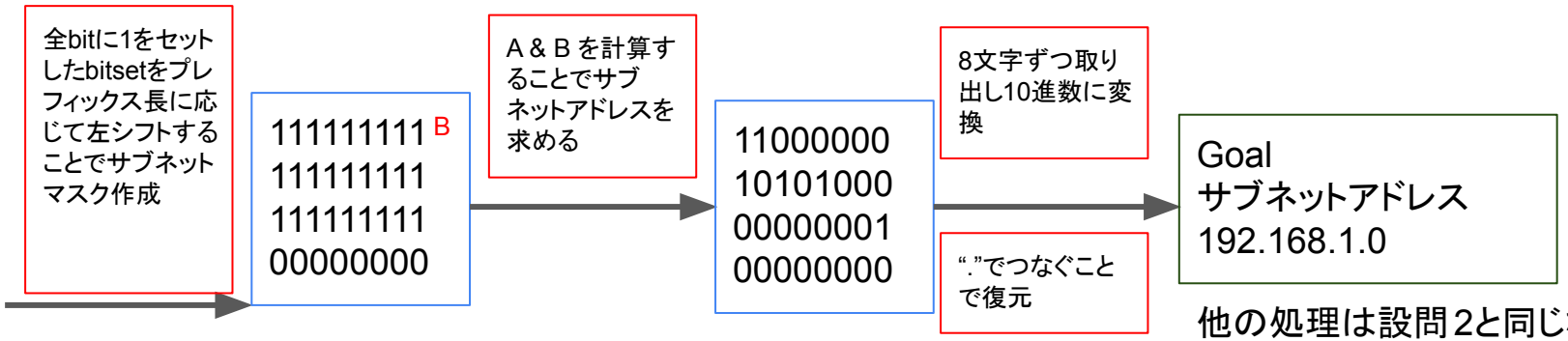
(同じサブネットのグループを管理するリストなどを作るべきだった？　)

# 6.設問4

## サブネットアドレスの作り方

使ったライブラリ : string, vector, bitset




他の処理は設問2と同じなため省略

# 6.テストデータ

## ・各テストデータと狙い

q1_in_1.txt  -> 一番シンプルなもの。タイムアウト後復旧したタイミングでログ出力できているか

q1_in_2.txt  -> 復旧してないパターン。今回は何も出力しないことを確認

q1_in_3.txt -> 故障->復旧->故障->復旧した場合ちゃんとlogを二回吐き出せているか

q2_in_1.txt -> q1_in_3.txtと同じデータ。N回以上の判定がちゃんとできているか確認

q2_in_2.txt -> 復旧していないものをちゃんと出せているか確認

q3_in_1.txt -> 過負荷期間が正しく出力されているか、平均応答時間の判定ロジックは正しいか

q3_in_2.txt -> 復旧していないものをちゃんと出力できるか、タイムアウトのデータを含めても大丈夫か

q4_in_1.txt -> シンプルなパターン。サブネットのグループ分けのロジックがちゃんと動いているか

q4_in_2.txt -> 故障->復旧->故障->復旧した場合や、復旧していない場合の出力がちゃんとできているか