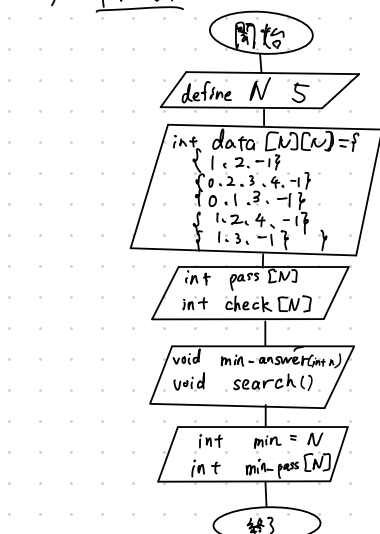


(1) 基本課題4

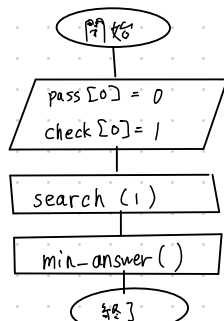
2164204-b4.c

経路探索の最適解を求めるプログラム

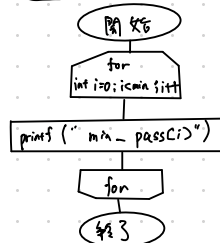
(2) 関数外



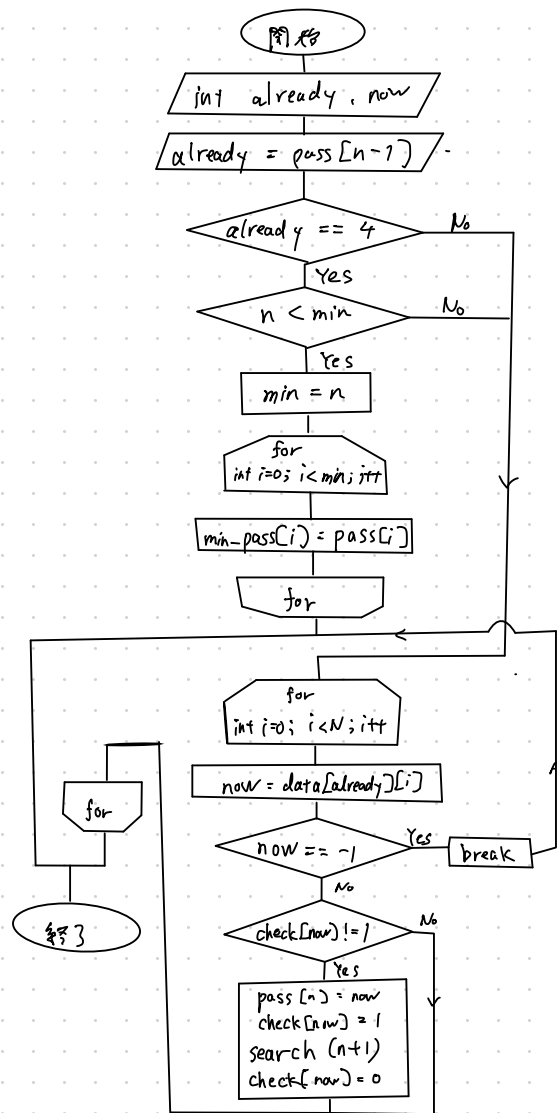
int main(void)



void min\_answer(void)



void search(int n)



(3)  $data[N][N]$  に関して、0行目では0地点から出ている

道の先の地点を返し、-1は0行目の終り点を表す。

$pass[N]$  で道順を記録。地点  $i$  を通ったら

$check[i] = 1$  とし、通っていない場合は  $check[i] = 0$ 。

main 関数で、最初0に始まるので  $pass[0] = 0$ 、

$check[0] = 1$ 。そして、 $pass[1]$  を探してみる。

$search(1)$  を呼び出す。already に5前に通った

$pass[1-1] = pass[0] (= 0)$ 。already が4でいい、

ゴールについていないので else 以下に入る。

$i = 0, 1, \dots, N$  について、 $now = data[already][i]$

( $= data[0][i]$ ) で現在いる地点から移動できる

場所を一つ選択。  $now = -1$ 、つまり行き先が

ない場合はストップ (break)。  $now \neq -1$  なら

$check[now] \neq 1$  (まだ通っていない) なら、

$pass[n] = now$ 、 $check[now] = 1$  とし

$now$  を通って行く、そして  $search(n+1)$  を呼び出す。

呼び出しが終了したら  $check[now] = 0$  とし

通っていないことにして他の道を試してみる。

呼び出しが終了、 $search(n)$  内の  $already = 4$  (ゴール

になっている)、手続  $n$  で現在までの最短手続  $min$

を比較し、 $n < min$  なら最短手続、最短経路を

更新する。main 関数で  $search$  関数が終わったら最短経路を返す。

(4)  $N \geq 5$  とする。  $data[N][N]$  に對して、  $i$  行目と  $i$  地点から出ている通の次の地点の重みとし、  
最後  $NULL$  代わりに  $-1$  とする。

地点:  $i$  通过, 那么  $i$  成分上 1. 通过, 2. (1) 1 块, 2 块 0 2 块  $check[N]$

経路探索関数 `search(int n)`

最短路径 存储空间优化) min-pass[N] 无定义

pass[0] = 0, check[0] = 1

第 2 次行程路程表

今更そう 2020.8.20 now 22.

already = 4 (2"-12) 以上、現在の年数 n が 最近年数 か  
判定し、最近年数以上更新する。

for  $x$  ( $i=0, \dots, N$ )  $\rightarrow$  now = data [already] ( $i$ )  $\geq c$ .

現在選抜している経路に  $W$  行,  $2$  行ない場合,

( $check[now] \neq 1$ )、現在選取している経路を通るかどうか ( $pass[n] = now, check[now] = 1$ )。

現在選取している経路の次の頂点を探すため、

$search(n+1)$  を再帰呼び出しする。

再帰呼び出しが終了した後、現在の経路を

通っていないことになるため、 $check[now] = 0$  にする。

(5) 今回は、最適解を求めるプログラムであった。

既述通り、再帰呼び出しを上手く使い、

コストもいつもより「分かりやすく」して、

理解が容易なように感じる。

前回はバックトラックがいまいち理解できていなかったが、

今回はしっかり理解できたと思う。