3-b 19257514 奈良　貴明　計算工学基礎レポート

メールに解説付きソースコードをcs3-b.cpp,list2.cpp,list2.hppとして添付しました.

・実行結果
添付したソースコードをコンパイルし実行すると以下の出力結果を得られる.

```
input map
 + + + + + + +
 + 0 s 0 0 0 +
 + 0 + + + + +
 + 0 0 0 0 + +
 + + + + 0 0 +
 + 0 0 0 0 g +
 + + + + + + +
queueinit
stackinit
start search(queue)
put:1,2   1,2 ->
get:1,2
put:1,3   1,3 ->
put:1,1   1,3 -> 1,1 ->
 + + + + + + +
 + 2 1 2 0 0 +
 + 0 + + + + +
 + 0 0 0 0 + +
 + + + + 0 0 +
 + 0 0 0 0 g +
 + + + + + + +
get:1,3   1,1 ->
put:1,4   1,1 -> 1,4 ->
 + + + + + + +
 + 2 1 2 3 0 +
 + 0 + + + + +
 + 0 0 0 0 + +
 + + + + 0 0 +
 + 0 0 0 0 g +
 + + + + + + +
get:1,1   1,4 ->
put:2,1   1,4 -> 2,1 ->
 + + + + + + +
 + 2 1 2 3 0 +
 + 3 + + + + +
 + 0 0 0 0 + +
 + + + + 0 0 +
 + 0 0 0 0 g +
 + + + + + + +
get:1,4   2,1 ->
put:1,5   2,1 -> 1,5 ->
 + + + + + + +
 + 2 1 2 3 4 +
```

```
 + 3 + + + + +
 + 0 0 0 0 + +
 + + + + 0 0 +
 + 0 0 0 0 g +
 + + + + + + +
get:2,1   1,5 ->
put:3,1   1,5 -> 3,1 ->
 + + + + + + +
 + 2 1 2 3 4 +
 + 3 + + + + +
 + 4 0 0 0 + +
 + + + + 0 0 +
 + 0 0 0 0 g +
 + + + + + + +
get:1,5   3,1 ->
 + + + + + + +
 + 2 1 2 3 4 +
 + 3 + + + + +
 + 4 0 0 0 + +
 + + + + 0 0 +
 + 0 0 0 0 g +
 + + + + + + +
get:3,1
put:3,2   3,2 ->
 + + + + + + +
 + 2 1 2 3 4 +
 + 3 + + + + +
 + 4 5 0 0 + +
 + + + + 0 0 +
 + 0 0 0 0 g +
 + + + + + + +
get:3,2
put:3,3   3,3 ->
 + + + + + + +
 + 2 1 2 3 4 +
 + 3 + + + + +
 + 4 5 6 0 + +
 + + + + 0 0 +
 + 0 0 0 0 g +
 + + + + + + +
get:3,3
put:3,4   3,4 ->
 + + + + + + +
 + 2 1 2 3 4 +
 + 3 + + + + +
 + 4 5 6 7 + +
 + + + + 0 0 +
 + 0 0 0 0 g +
 + + + + + + +
get:3,4
```

```
put:4,4   4,4 ->
 + + + + + + +
 + 2 1 2 3 4 +
 + 3 + + + + +
 + 4 5 6 7 + +
 + + + + 8 0 +
 + 0 0 0 0 g +
 + + + + + + +
get:4,4
put:4,5   4,5 ->
put:5,4   4,5 -> 5,4 ->
 + + + + + + +
 + 2 1 2 3 4 +
 + 3 + + + + +
 + 4 5 6 7 + +
 + + + + 8 9 +
 + 0 0 0 9 g +
 + + + + + + +
get:4,5   5,4 ->
5,5 is GOALLLLL!!!!!!
finish search(queue) and find shortest path
show result
 + + + + + + +
 + 2 1 0 0 0 +
 + 3 + + + + +
 + 4 5 6 7 + +
 + + + + 8 9 +
 + 0 0 0 0 10 +
 + + + + + + +
start search(stack)
push:1,2   1,2 ->
push:1,3   1,2 -> 1,3 ->
 + + + + + + +
 + 0 1 2 0 0 +
 + 0 + + + + +
 + 0 0 0 0 + +
 + + + + 0 0 +
 + 0 0 0 0 g +
 + + + + + + +
push:1,4   1,2 -> 1,3 -> 1,4 ->
 + + + + + + +
 + 0 1 2 3 0 +
 + 0 + + + + +
 + 0 0 0 0 + +
 + + + + 0 0 +
 + 0 0 0 0 g +
 + + + + + + +
push:1,5   1,2 -> 1,3 -> 1,4 -> 1,5 ->
 + + + + + + +
 + 0 1 2 3 4 +
```

```
 + 0 + + + + +
 + 0 0 0 0 + +
 + + + + 0 0 +
 + 0 0 0 0 g +
 + + + + + + +
pop:1,5   1,2 -> 1,3 -> 1,4 ->
pop:1,4   1,2 -> 1,3 ->
pop:1,3   1,2 ->
pop:1,2
push:1,1   1,1 ->
 + + + + + + +
 + 2 1 2 3 4 +
 + 0 + + + + +
 + 0 0 0 0 + +
 + + + + 0 0 +
 + 0 0 0 0 g +
 + + + + + + +
push:2,1   1,1 -> 2,1 ->
 + + + + + + +
 + 2 1 2 3 4 +
 + 3 + + + + +
 + 0 0 0 0 + +
 + + + + 0 0 +
 + 0 0 0 0 g +
 + + + + + + +
push:3,1   1,1 -> 2,1 -> 3,1 ->
 + + + + + + +
 + 2 1 2 3 4 +
 + 3 + + + + +
 + 4 0 0 0 + +
 + + + + 0 0 +
 + 0 0 0 0 g +
 + + + + + + +
push:3,2   1,1 -> 2,1 -> 3,1 -> 3,2 ->
 + + + + + + +
 + 2 1 2 3 4 +
 + 3 + + + + +
 + 4 5 0 0 + +
 + + + + 0 0 +
 + 0 0 0 0 g +
 + + + + + + +
push:3,3   1,1 -> 2,1 -> 3,1 -> 3,2 -> 3,3 ->
 + + + + + + +
 + 2 1 2 3 4 +
 + 3 + + + + +
 + 4 5 6 0 + +
 + + + + 0 0 +
 + 0 0 0 0 g +
 + + + + + + +
push:3,4   1,1 -> 2,1 -> 3,1 -> 3,2 -> 3,3 -> 3,4 ->
```

```
 + + + + + + +
 + 2 1 2 3 4 +
 + 3 + + + + +
 + 4 5 6 7 + +
 + + + + 0 0 +
 + 0 0 0 0 g +
 + + + + + + +
push:4,4   1,1 -> 2,1 -> 3,1 -> 3,2 -> 3,3 -> 3,4 -> 4,4 ->
 + + + + + + +
 + 2 1 2 3 4 +
 + 3 + + + + +
 + 4 5 6 7 + +
 + + + + 8 0 +
 + 0 0 0 0 g +
 + + + + + + +
push:4,5   1,1 -> 2,1 -> 3,1 -> 3,2 -> 3,3 -> 3,4 -> 4,4 -> 4,5 ->
 + + + + + + +
 + 2 1 2 3 4 +
 + 3 + + + + +
 + 4 5 6 7 + +
 + + + + 8 9 +
 + 0 0 0 0 g +
 + + + + + + +
push:5,5   1,1 -> 2,1 -> 3,1 -> 3,2 -> 3,3 -> 3,4 -> 4,4 -> 4,5 -> 5,5 ->
5,5 is GOALLLLL!!!!!!
show result
 + + + + + + +
 + 2 1 2 3 4 +
 + 3 + + + + +
 + 4 5 6 7 + +
 + + + + 8 9 +
 + 0 0 0 0 10 +
 + + + + + + +
```

実行結果から，キューを用いた最短経路探索では，移動可能なところを並行して探索していることが確認できる．一方スタックを用いた探索では，一つのパスを探索していき行き止まりになったら引き返して別のパスを探索していることが確認できる．前者を幅優先探索といい，後者を深さ優先探索という．

・考察感想

キューを用いた最短経路探索のアルゴリズムは，注目座標の縦横4近傍に対して移動可能かを判別し，移動可能であればその座標をキューにputする->moveCheck_queue関数
4近傍の探索が終わるとキューからputしていた座標をgetし，その座標に対してまた4近傍で移動可能かを判別することを繰り返していく．キューは先に入れたデータを取り出すことができるので同じスッテプ数のところから探索済みになるため，ゴールを見つけたときのステップ数がゴールまでの最小距離になる．

　スタックを用いた経路探索のアルゴリズムは，注目座標の縦横4近傍に対して，移動可能かを判別するのは同様だが，移動可能な場所が見つかった瞬間にその座標をスタックにpushし，移動する点が大きく違う．これにより，スタックを用いた探査アルゴリズムは行き止まりに当たるまでステップを繰り返す．そして，行き止まりに当たった場合，未探索の移動可能座標が見つかるまで，スタックに格納されていた座標を遡っていく．これにより，スタックを用いいた経路探索では，必ずしも最短経路が見つかるとは限らない.

　今回のプログラムで工夫した点としては，char型の迷路は人間が入力したり確認したり使用するには便利だが，プログラム的に処理するには不向きだったため，int型の迷路に変換し処理したところである．これにより，数値的な処理が容易になりプログラムの可読性も向上した.
　この次の最長経路探索のアルゴリズムがまだ思いつかないのではやめに考えつきたい.
　今回はc++を使用したが，c寄りの書き方をしてしまったためあまりc++の利点を活かせなかったので次回は，c++の利点を活かせるようにしたい.

・協力者
19257504　太田那菜
・参考サイト
迷路探索プログラムのアルゴリズム 2019/12/13
”https://proglight.jimdo.com/programs/vba/maze/”