Queue	Stack	OrderedCollection
+enqueue	push	offer
-dequeue	pop	take
✓ is_empty	isempty	isempty

IP offer N things and take N things

get all offers

Dequeue gives back in order enqueue

pop gives last push

enq Bill
enq Ann
deq → Bill

enq Ada
deq → Ann

$1+2+\cdots n$

$\frac{n(n+1)}{2}$ is $O(n^2)$

Stack	SLL	AL
push	O(1)	Amortized O(1)
pop	O(1)	Amortized O(1)
	(insert front)	(insert back)

Queue	SLL	SLL w/T	AL	CAL
enqueue	O(n)	O(1)	O(1)	Amortized O(1)
dequeue	O(1)	O(1) back	O(n) back	Amortized O(1)

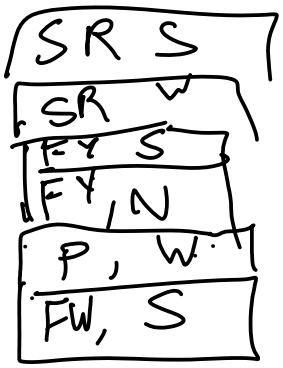
Place, Direction

```
class Task {
public:
    ....
private:
    string place
    string direction
}
```

Stack<Task>

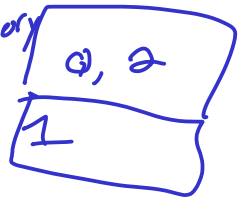
get to a room:
    create task for each direction
    add all of them to DS

- add tasks for 1st location
  as long as there are tasks:
    take 1st task
    go where it says ——— If I found it, I'm done!
    if haven't been here
        add new tasks for this loc



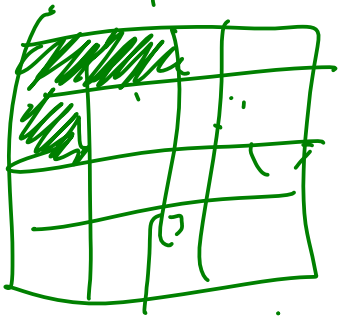
FW, P, FG, FY
SR

- Can be lucky
- Can be unlucky
- Might not finish
- Less memory



Depth-first search

- Can't really be (un)lucky
- Will always find the answer*



Breadth-first search