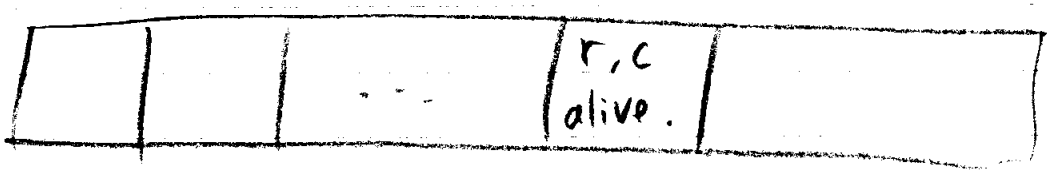
GRIDWORLD - APPROACH - ZERO

MAIN DATA STRUCTURE:

ARRAY/VECTOR OF "Person" structures/objects

INDEXED BY ID

```
Person {
    int r, c;
    bool alive;
}
```



id

birth ~~$\Theta(N_{total})$~~ $\Omega(N_{current})$
death $O(1)$
whereis $O(1)$
move $O(1)$
members $\Theta(N_{total})$

pop rc $\Theta(N_{total})$
pop $\Theta(N_{total})$

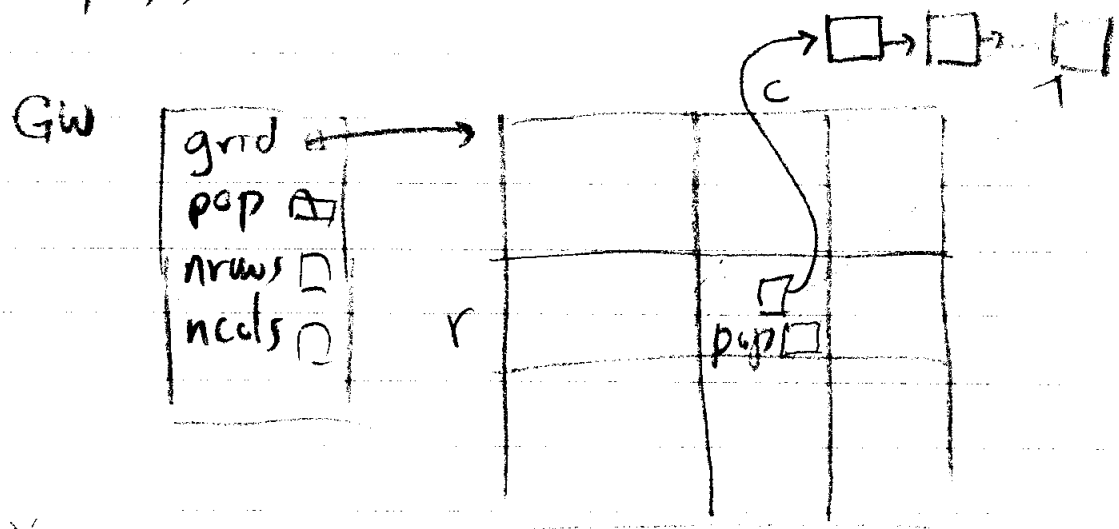
APPROACH - 1

IDEA: Make members faster by having a 2D array representing the world instead of 1D array of people. Fix pop too!

District {

List members; // ids of members in order of servants
int pop;

}



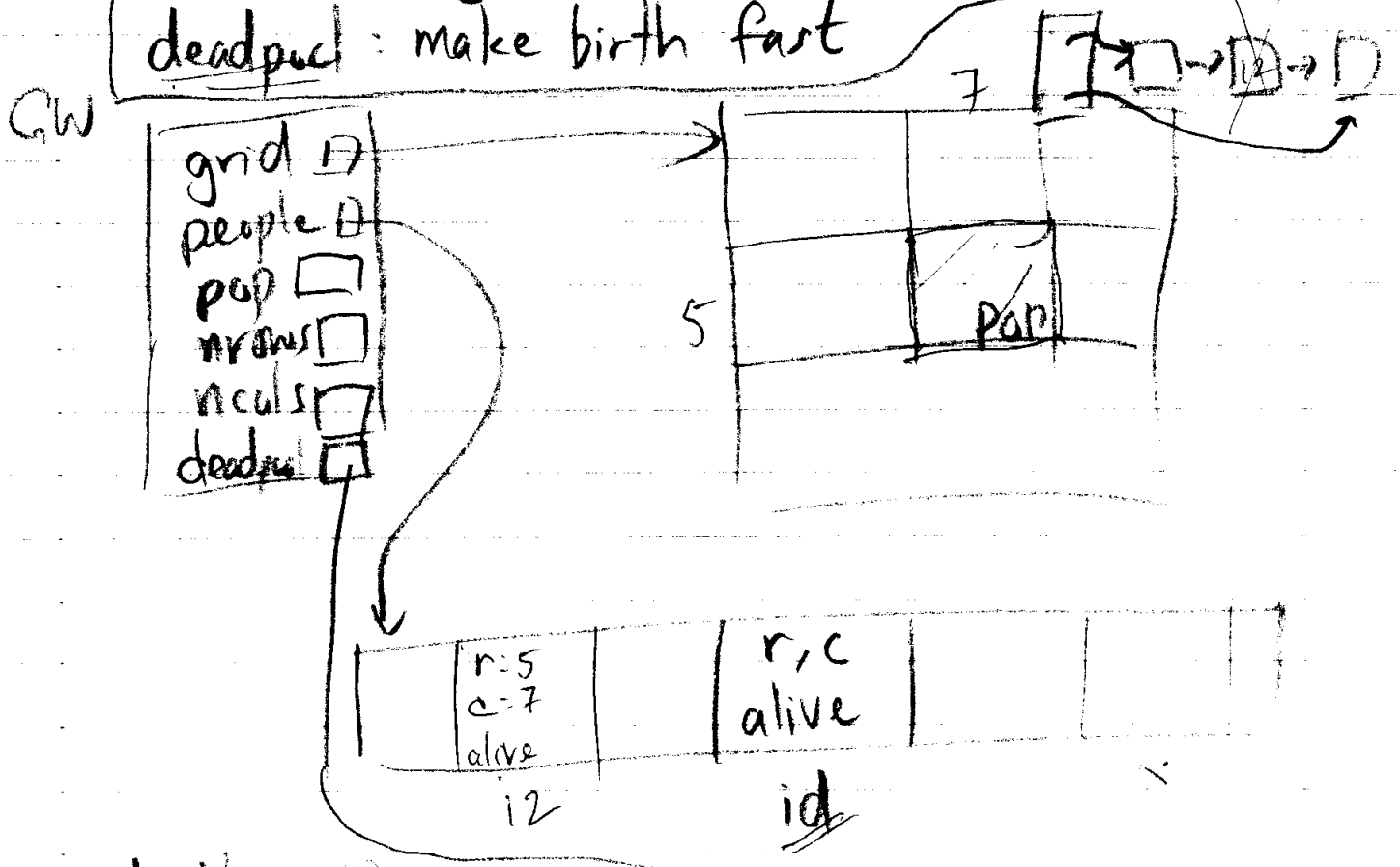
birth	YUCK
death	BAD
where	$O(RC + N_{living})$
move	BAD
members	$\Theta(N_{rc})$
pop r c	$O(1)$
pop	$O(1)$

APPROACH 2 - COMBINE 0 + 1 + Deadpool

people array: make where is fast

District grid: make members fast

deadpool: make birth fast



birth $O(1)$

death $\Theta(N_r o c_c)$

whereis $O(1)$

move $\Theta(N_r o c_c)$

members $\Theta(N_{rc})$

pop r c $O(1)$

pop $O(1)$

APPROACH 3

Get move & death down to O(1)