# Problem Set 5

Problem A

Jumping on walls was maybe the fastest algorithm I ever got in this course! I think the question is pretty intuitively linked to shortest paths, I pretty quickly identified that. Actually looking at it again, I'm pretty sure it wasn't even a shortest path question, but just a basic BFS/DFS question.

The methodology I applied was just doing a BFS on locations that the ninja can be. Initially, for simplicity's sake, I coded this up without taking into consideration the water, and then would add the water later. That way, each location on the wall was a node on the graph, and each had edges to possible next locations, (jump, step up, step down). Edges were calculated on the fly in order to save memory. They would first start on the LHS wall, at height = 1, and then have two edges to different locations. BUT, the spikes would invalidate some edges, so I created some bool's in order to check if each step was valid before adding it to the queue. If we reach a location that is higher, then we win.

Then to add water, I just changed the locations from pairs of ints, to a struct, representing a 'state' instead. This is because the Ninja can be at the same location, with different water heights, so we would need to know where the water is at each 'state' to know if it would invalidate the jump. Then a check to see if the current height was less than or equal the water was added in to skip over some states.

My brain for this question initially went to BFS, but I also then realised DFS would work as well after chatting to a friend about this. This was confirmed by just changing the queue to a stack, and it all still worked!