Vectors from scratch.

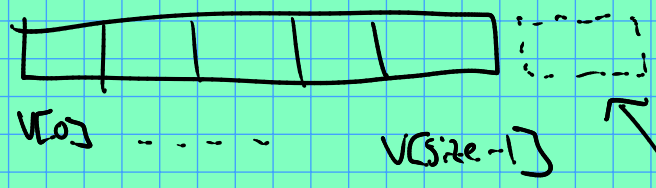
How to represent the data?

Need to store

size

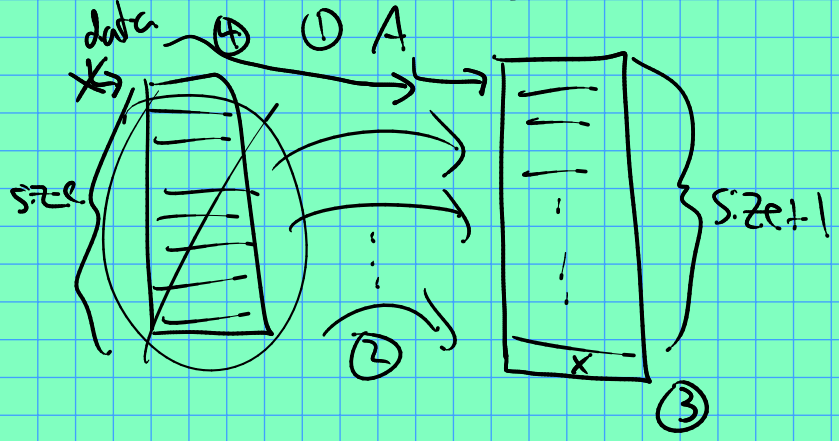
data (for v[0], v[1] ...)

2 items above could be enough, but it won't be ideal for performance. Let's see why: how to implement push_back(x)?



might be used already!

In general, we might have to find somewhere else to store the data (now size+1 elements).



in C++:

```
int* A = new int[size+1]; // ①
for (int i=0; i<size; i++)
    A[i] = data[i];       // ②
A[size++] = x;            // ③
delete[] data;
v.data = A;               // ④
```

What would be the cost of doing n consecutive push_back's?

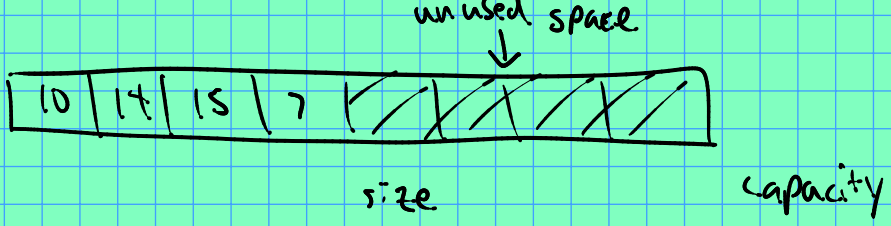
If size $= n$, one cost $\approx n$.

So total would be

$\approx 1 + 2 + 3 + \cdots + n \approx n^2$.

$\therefore$

We could instead be more generous:



capacity $\equiv$ length of data array (including unused space)

size $\equiv$ # valid elements, which live in data[0... size-1].

Exercise: show that by doubling the capacity when we run out of space, cost of n push_backs $\approx cn$ for constant c (e.g. $c = 2$).

Furthermore, convince yourself that doing capacity += c for any constant c will still give $\approx n^2$ steps.