Exercise: given 2 <u>sorted</u> vectors, produce a new vector that is also sorted, and has all elements of both input vectors.

E.g. if $V1 = 2\ 3\ 5\ 9\ 13$

$V2 = 1\ 4\ 8\ 9$

the new vector should be $V = 1\ 2\ 3\ 4\ 5\ 8\ 9\ 9\ 13$

$$V1 = 2\ 3\ \overset{i1}{\overline{5}}\ 9\ 13$$

$$V2 = 1\ 4\ \underset{i2}{\underline{8}}\ 9$$

$$V = 1\ 2\ 3\ 4\ \ldots$$

$$O(n)$$

$$= O(|V1| + |V2|)$$

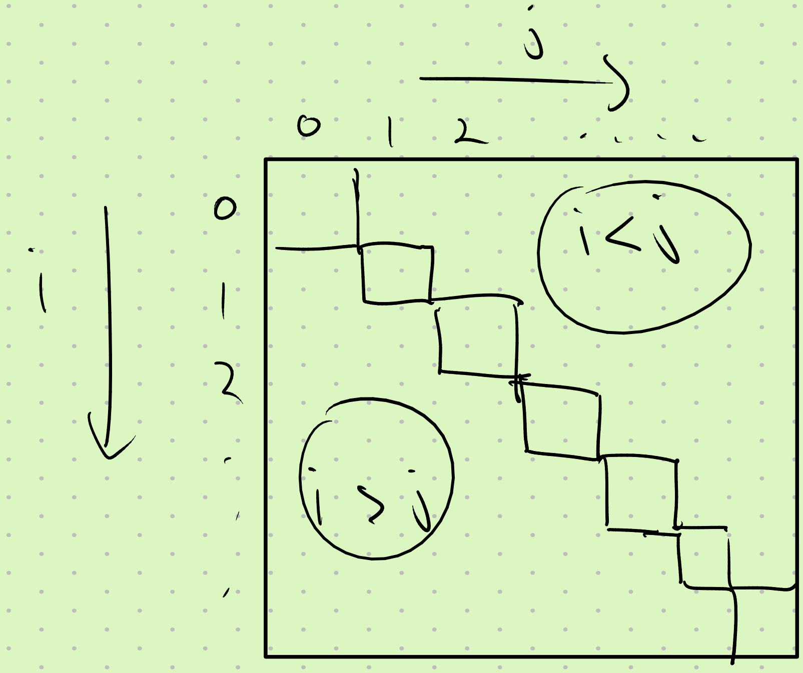
Details?

```
size_t i1 = 0, i2 = 0;   // candidate locations

while (i1 < V1.size() && i2 < V2.size()) {
    if (V1[i1] < V2[i2])
        V.push_back(V1[i1++]);
    else V.push_back(V2[i2++]);
}
```

```
while (i1 < V1.size())  V.push_back(V1[i1++]);
while (i2 < V2.size())  V.push_back(V2[i2++]);
```

From last time: looking for 2 vector elements that sum to a target t: i ≠ j s.t. V[i] + V[j] == t.



(say n = V.size())

Want to go through all (i,j) pairs with i>j.

```
for (i=1;  i<n;  i++)  {
      for (j=0; j<i  j++)  {
          if (V[i]+V[j] == t) ....
      }
}
```

iterates through all i>j

Iterate through all i<j

```
for (i=0;  i<n-1;  i++) {
      for (j=i+1; j<n; j++) {
          ....
```