# Searching in a List (Q)

- Search for a target **x** in a list **v**
- Start at index 0, keep checking *until* you find it or *until no more element to check*

```
     0    1    ...  k  ...
v  | 12 | 35 | 33 | 15 | 42 |

x  | 14 |
```

**Linear search**

Suppose another list is twice as long as v. The expected "effort" required to do a linear search is

A. Squared
B. Doubled
C. The same
D. Halved
E. I don't know

See **search.py**    7

---

# Searching in a List (A)

- Search for a target **x** in a list **v**
- Start at index 0, keep checking *until* you find it or *until no more element to check*

```
     0    1    ...  k  ...
v  | 12 | 35 | 33 | 15 | 42 |

x  | 14 |
```

**Linear search**

Suppose another list is twice as long as v. The expected "effort" required to do a linear search is

A. Squared
B. Doubled    **CORRECT**
C. The same
D. Halved
E. I don't know

Effort is *linearly* proportional to list size. Needs $n$ comparisons for list of size $n$ (at worst case). [8]