The set data structure.

vector vs set :

vector	set
arbitrary storage	same.
push_back(x)	insert(x)
can store duplicates	No duplicates
"random access" (v[i])	No such equivalant
cost for search ≈ n steps	cost for search: ≈ log n steps.

Remarks: they model mathematical sets $S \subseteq U$ (U is some universe).

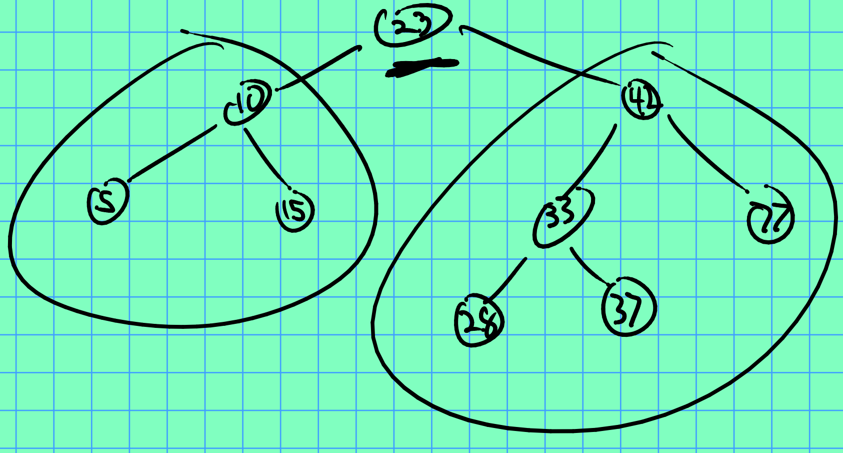
$$f_S : U \to \{0,1\} \equiv S \subseteq U$$

$$f_S(x) = 1 \iff x \in S$$

so, $S = f_S^{-1}(\{1\})$.

f is called the "characteristic function" of S.

Sets behind the scenes:



Note: when searching through a vector, each comparison rules out only *one* location.

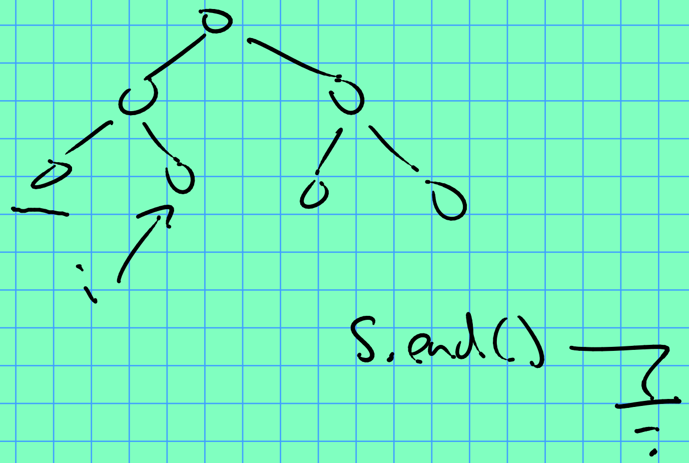
In a set, each comparison rules out *half* of what's left!

Hence the $\approx \log_2 n$ steps for search:

$$\frac{n}{2^k} = 1 \Rightarrow n \approx 2^k$$

$$= k \approx \log_2 n.$$

"*iterators*" (kind of like pointers...)



They provide a pointer-like interface, but the implementation might be more complex behind the scenes.