Recherche dichotomique

1. Recherche séquentielle
   - examiner chaque élément
   - pire cas : $O(n)$
2. Condition
   - liste triée
3. Principe
   - comparer à l'élément du milieu
   - égal : trouvé
   - plus petit : moitié gauche
   - plus grand : moitié droite
4. Python
   - indices `gauche, droite`
   - `milieu = (gauche + droite) // 2`
   - renvoie un booléen ou l'indice
5. Complexité
   - un million : 20 comparaisons
   - $O(\log n)$
   - taille divisée par deux
6. Terminaison
   - variant : `droite - gauche`