Search

Assume we are given three procedures:

- `Search` – which performs a straight forward search
- `BinarySearch` – which performs a binary search
- `Sort` – sorts in $\mathcal{O}(n \cdot \log(n))$

The procedures below take an integer array `A` of size n and an array `B` of size k as input. All of the procedures should print the numbers which are contained in both `A` and `B`.

```
1 Procedure Intersection1(A, B)
2 |   for x in B do
3 |   |   if Search(A, x) then
4 |   |   |   print(x)
```

```
1 Procedure Intersection2(A, B)
2 |   for x in B do
3 |   |   if BinarySearch(A, x) then
4 |   |   |   print(x)
```

```
1 Procedure Intersection3(A, B)
2 |   HeapSort(A)
3 |   for x in B do
4 |   |   if BinarySearch(A, x) then
5 |   |   |   print(x)
```

```
1 Procedure Intersection4(A, B)
2 |   for x in B do
3 |   |   HeapSort(A)
4 |   |   if BinarySearch(A, x) then
5 |   |   |   print(x)
```

Three of these procedures always give the correct result, whereas one may give an incorrect result. We want to know which variant to use depending on the size of k (i.e. the length of `B`) in relation to n (i.e. the length of `A`).