

از آنجایی که اعضای A در محدوده 1 تا $2n$ هستند، میتوان آرایه‌ای به انداز $2n$ ایجاد کرد و اعداد را براساس اندازه در این آرایه ذخیره کرد. بدین صورت که اگر عنصر i ام این آرایه 1 باشد آنگاه این عدد در آرایه A وجود دارد. پس از ذخیره کردن A در این آرایه با بررسی اعضای B یکسان بودن یا نبودن این آرایه‌ها را بررسی می‌کنیم.

```
def are_equal(A: list[int], B: list[int]):
    n = len(A)
    helper_list = [0] * (2 * n + 1)
    for number in A:
        helper_list[number] = 1
    for number in B:
        if helper_list[number] == 0:
            return False
    return True
```

در این تابع پس از ذخیره اطلاعات در `helper_list` اگر عددی در B وجود داشته باشد که مقدار آن در `helper_list` 1 نباشد آنگاه این دو آرایه یکسان نیستند و اگر چنین عددی وجود نداشت باهم برابرند.

اگر عملیات پایه‌ای را انتصاب در نظر بگیریم، آنگاه تعداد عملیات‌های پایه‌ای تنها به اندازه A وابسته خواهد بود و برابر اندازه A خواهد بود. اگر عملیات پایه‌ای را مقایسه در نظر بگیریم آنگاه تنها به اندازه B وابسته خواهد بود و در بدترین حالت برابر با اندازه B خواهد بود اما اگر A و B به صورت تصادفی انتخاب شده باشند به صورت میانگین بین ۱ تا ۳ مقایسه برای نشان دادن اینکه یکسان نیستند کافی است زیرا احتمال برابر بود آنها بسیار کم است و اگر عملیات پایه‌ای را دسترسی به عنصری در هریک از آرایه‌ها در نظر بگیریم، آنگاه تابع نشان‌دهنده تعداد عملیات‌های پایه‌ای به صورت $T(n) = n + k$ به طوری که k تعداد مقایسه‌های ما خواهد بود که دسترسی به عناصر B را نشان می‌دهد: در زیر نمونه‌ای از خروجی برنامه برای مجموعه‌های تصادفی A و B آورده شده است:

length of A and B is 10^1
A and B are not equal
a_access: 10, b_access: 2, assigns: 10
helper_access: 2, comparisons: 2

length of A and B is 10^2
A and B are not equal
a_access: 100, b_access: 1, assigns: 100
helper_access: 1, comparisons: 1

length of A and B is 10^3
A and B are not equal
a_access: 1000, b_access: 2, assigns: 1000
helper_access: 2, comparisons: 2

length of A and B is 10^4
A and B are not equal
a_access: 10000, b_access: 2, assigns: 10000
helper_access: 2, comparisons: 2

length of A and B is 10^5
A and B are not equal
a_access: 100000, b_access: 1, assigns: 100000
helper_access: 1, comparisons: 1

length of A and B is 10^6
A and B are not equal
a_access: 1000000, b_access: 1, assigns: 1000000
helper_access: 1, comparisons: 1