به نام خداوند هستی‌بخش

ساختمان داده و الگوریتم - تمرین پنجم

امیر محمد کریمی

amir.karimi6610@gmail.com

تاریخ تحویل : ۶ خرداد

۱. مجموعه A شامل n عدد حقیقی $x_1, x_2, ..., x_n$ میباشد. الگوریتمی با هزینه زمانی $O(n \log n)$ ارائه دهید که نشان دهد آیا دو عدد در این مجموعه وجود دارد که مجموعشان عدد مشخص P شود.

۲. دو آرایه $X[1..n]$ و $Y[1..m]$ داریم که $n < m$ است. فرض کنید X مرتب نباشد اما Y به صورت صعودی مرتب شده باشد. الگوریتمی با هزینه زمانی متوسط $O(n)$ ارائه دهید که مقدار k امین کوچکترین عدد (یعنی عددی که دقیقا از $k-1$ عدد دیگر بزرگتر باشد) را در مجموعه $X \bigcup Y$ بیابد. (فرض کنید چنین عددی با شرط گفته شده وجود دارد)

۳. آرایه نامرتب P به طول n داده شده است. الگوریتمی در زمان $O(n \log n)$ ارائه دهید به طوری که خروجی آن آرایه Q باشد. هر عضو این آرایه دارای شرط زیر است :

$$Q[i] = (\text{تعداد } j \text{ هایی که } j > i \text{ و } P[j] < P[i])$$

۴. الگوریتم مرتب سازی دوطرفه به صورت زیر تعریف شده است:

```
function mySort(A[]){
 do{
   swapped := false;
   foreach i in 0 to length(A)-2 do{
     if A[i] > A[i+1] then{
       swap(A[i],A[i+1]);
       swapped := true;
     }
   }
   if swapped == false then
     break;
   swapped := false;
   foreach i in length(A)-2 down to 0 do{
     if A[i] > A[i+1] then{
       swap(A[i],A[i+1])
       swapped := true;
```

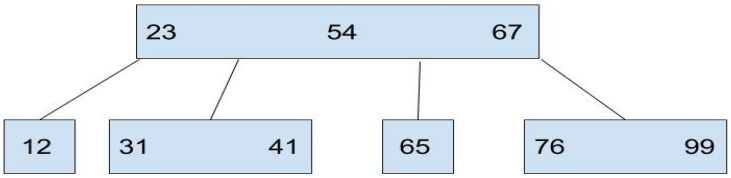
```
        }
      }
    }while swapped;
}
```

در حقیقت این روش مرتب سازی، تغییر یافته ی الگوریتم $bubble\ sort$ میباشد بدین صورت که علاوه بر پیمایش آرایه از ابتدا تا انتها برای انتقال دادن بزرگترین عضو به انتهای آرایه، پیمایش دیگری نیز از انتهای آرایه به سمت ابتدای آن صورت میگیرد که کوچکترین عضو را به ابتدای آرایه انتقال میدهد. در نهایت، در صورتی که در یک پیمایش مکان عضوی از آرایه تغییر نکند، یعنی آرایه مرتب شده است.

(آ) پیچیدگی زمانی الگوریتم فوق را به دست آورید.

(ب) ویژگی های رو به رو را برای الگوریتم مرتب سازی فوق بررسی کنید. (درجا، مقایسه ای، پایداری)

۵. در درخت $B-Tree$ زیر با $t=2$ ابتدا عدد 80 و سپس در درخت حاصل، عدد 75 را $insert$ کنید.



۶. یک جدول $hash$ با 8 خانه داریم.

(آ) اگر از روش $double\ hashing$ برای رفع برخورد استفاده کنیم، با استفاده از توابع داده شده، اعداد ۲۱، ۳۴، ۵۳، ۱۷، ۷۸، ۳۵ و ۴۱ را به ترتیب از راست به چپ در جدول قرار دهید.

$$h_1(k) = k \bmod 8\ ,\ h_2(k) = 3 + 2 * (k \bmod 5)$$

(یادآوری: برای قرار دادن یک عدد در جدول $hash$ به روش $double\ hashing$ از تابع $H(k,i) = (h_1(k) + i * h_2(k))\ \ mod\ TableSize$ استفاده میکنیم.)

(ب) تابع h_2 را به گونه ای تغییر دهید که هنگام اضافه کردن داده های مذکور، تعداد برخورد برای هر عدد حداکثر ۱ باشد.

(ج) اگر از روش $chaining$ برای رفع برخورد استفاده کنیم به طوری که احتمال قرار گرفتن هر عضو در خانه های جدول برابر باشد، احتمال آن که در فرآیند اضافه کردن 7 عدد به جدول، دقیقا 6 کلید در یک خانه قرار گیرند را محاسبه کنید.

۷. (امتیازی) فرض کنید n عدد صحیح داریم که تعداد تکرار آنها بسیار زیاد است؛ به طوری که حداکثر $O(log\ n)$ عدد متفاوت در میان این اعداد وجود دارد. با توجه به این شرایط الگوریتمی طراحی کنید که با هزینه ای کمتر از $O(n \log n)$ این اعداد را مرتب کند.