

# আসিফের হ-য-ব-র-ল

জানি জানার শেষ নাই, তবু শুরু করতে দোষ কোথায়??

## Painless Binary Search

Published on June 20, 2014 | Time 18:00

এত সহজভাবে নিয়োগ না ব্যাপারটা। বাইনারী সার্চ যতটা সহজ ভাবছে ততটা সহজ আসলে এটা না, ভিভাগ করতে গেলে জান বের হয়ে যাওয়ার উপক্রম হয় মাঝে মাঝে সবারই। এখনও ৯০% কোডাররা এটা on sight কোড করতে গেলে বিভিন্ন সমস্যা পড়ে যায়। এতক্ষণে হয়তো তোমারও মনে পড়ে যাওয়ার কথা যে বাইনারী সার্চ কোড করতে তোমারও কতবার গুগল করে বা বই দেখে ঠিক করে নেয়া লেগেছে কোডটা। তারপরও বাগ থেকে যেতো, ইনফিনিটি লুপ অথবা ভুল উত্তর। বাইনারী সার্চ কোড করতে গেলে যে ধরনের বাগে আমরা পড়ি তার জন্য একটা উদাহরণ দিয়ে দি আগে:

ধরো তোমার কাছে একটা আরে আছে:

```
[1, 1, 3, 3, 4, 4, 4, 4, 5, 9, 20, 20, 21, 21, 21]
```

এখন তুমি সার্চ করতে চাচ্ছে ৪ এর সর্বোনিম্ন অবস্থানটা, অর্থাৎ ৪ নং ইনডেক্স।

তুমি কোড করলা:

```
int bug(int Q, int n){
    int lo=0, hi=n, mid;
    while(lo<hi){
        int mid=(lo+hi)/2;
        if(arr[mid]>=Q){
            hi=mid-1;
        } else lo=mid+1;
    }
    return mid;
}
```

এখানে  $Q=4$  দিলা। রান করে দেখো কি সব উল্টাপাল্টা উত্তর আসছে। এখন তুমি বসে বসে কোড ঠিক করার চেষ্টা করলে।  $while(lo<hi)$  হবে নাকি  $while(lo<=hi)$  হবে?  $lo=mid$  দিলে কি কাজ হবে?  $return$  কি  $hi/lo/mid$  কোনটা করবো?  $hi=mid-1$  দেয়ার জন্য কি সমস্যা হচ্ছে? এসব করতে শেষ পর্যন্ত গুগল আমার হেল্প নিলা অথবা বইয়ের সূত্রে কোডটা দেখে ঠিক করে নেয়ার ট্রাই করলা। সচরাচর এমন সমস্যা প্রায় সব কোডারেরই হয়। যাইহোক, এই আর্টিকেলটা লিখতেছি কেবল বাইনারী সার্চের এই অসুখ সময় অগচয় হ্রাস করার জন্য। আর্টিকেলটার শেষটুকু পড়ার পর আশা করি তোমাদের বাইনারী সার্চ কোড করার সম্পূর্ণ আত্মবিশ্বাস+নির্ভুল কোড করার সামর্থ্য চলে আসবে।

প্রথমে বলে নি বাইনারী সার্চ দুই ধরনেরই হয়:

- আমরা যে আইটেমটা সার্চ করতে চাই সেটার lowest position টা

নাইলে

- আমরা যে আইটেমটা সার্চ করতে চাই সেটার highest position টা

## Lower Bound Searching

এখানে আমরা চাই আরেতে আইটেমটার সর্বোনিম্ন পজিশনটা খুঁজে বের করতে। অর্থাৎ  $p(f(x))$  টা  $x$  এর সর্বোনিম্ন যে মানের জন্য সত্য হবে। উপরে যে আরেটা দিয়েছিলাম ওটাতে যদি ৪ খুঁজতে চাই তাহলে আমরা ৪ নং ইনডেক্সটা রিটার্ন করবো। তোমাদেরকে একটা জিনিষ উপহার দিলাম: `best_so_far` . কোড দেখা শেষ হলে আমাকে মনে মনে গালি দিয়ে না কারন এটা হয়তো অ্যালগরিদমের বইতে দেয়া নাই, সামান্য একটু আপডেট।

```
int LowerBound(int Q, int n) {
    int lo=0, hi=n-1;
    int best_so_far=-1;
    while(lo<=hi) {
        int mid=(lo+hi)/2;
        if(arr[mid]>=Q) {
            if(arr[mid]==Q){
                best_so_far=mid;
            }
            hi=mid-1;
        } else lo=mid+1;
    }
    return best_so_far;
}
```

যেহেতু আমরা lower bound খুঁজে বের করতে চাচ্ছি তাই  $arr[mid]>=Q$  খুঁজছি যেটা আমাদের সর্বনিম্ন ইনডেক্সের দিকে নিয়ে যাবে, আর যদি  $arr[mid]==Q$  হয় তাহলেই `best_so_far` আপডেট করবো। কোড দেখা শেষ হলে upper bound এর কোডটা দেখে নাও তাহলে সম্পূর্ণ ক্রিয়ার হয়ে যাবে ধারণা।

## Upper Bound Searching

এখানে তোমাকে আইটেমটার সর্বোচ্চ পজিশনটা খুঁজে বের করতে হবে অর্থাৎ  $p(f(x))$  সত্য হবে সর্বোচ্চ  $x$  এর যে মানের জন্য। কোডটা দেখো এবার:

```
int UpperBound(int Q, int n) {
    int lo=0, hi=n-1;
    int best_so_far=-1;
    while(lo<=hi) {
        int mid=(lo+hi)/2;
        if(arr[mid]<=Q) {
            if(arr[mid]==Q){
                best_so_far=mid;
            }
            lo=mid+1;
        } else hi=mid-1;
    }
    return best_so_far;
}
```

ছোট্ট করে আমার সম্পর্কে...



আমি (আবু আসিফ খান চৌধুরী), বর্তমানে Secure Link Services | SELISE Rockin' Software এ Software Engineer হিসেবে কাজ করছি। পড়াশুনা করেছি Rajshahi University of Engineering and Technology থেকে। প্রোগ্রামিং ও অ্যালগরিদমের প্রতি মারাত্মক আকর্ষণ এবং নতুন কিছু শেয়ার করার জন্য এখানে লেখালিখি করি এবং করবো। রুগটাতে আমি কনটেন্ট প্রোগ্রামিং, অ্যালগরিদম, এবং এ বিষয়ক আমার জানা তথ্যগুলো সহজ ভাষায় লেখার চেষ্টা করবো।

আমাকে ফলো করতে পারেন

[গিটহাব](#) [গুগল+](#) [লিঙ্কডইন](#)

মোট পেজভিউ

036196

```
    } else m=mid-1;
  }
  return best_so_far;
}
```

দুইটা কোড কম্পেয়ার করলে বুঝতে পারবে trick টা কেবল `arr[mid]<=Q` এবং `arr[mid]<=Q` তেই আছে জাস্ট `lo=mid+1` `hi=mid-1` এই দুইটার পজিশন চেন্স করেছি UpperBound ফাংশনে।

হ্যা, বাইনারী সার্চ এতটাই সিম্পল। আশা করি এবার থেকে এটা কোড করতে ঝামেলা হবে না, হইলে লেখক দায়ী নয়। ;)

Keep coding... :)