Day 4

1.in binary search we usually prefer to do beg + (end-beg)/2 so as to avoid overflow as any value to be represented as mid will always lie less than (end-beg)

2.

|  |
| --- |
| #include <iostream> |
|  |  |
|  | using namespace std; |
|  | int ternary(int beg, int end, int n, int ar[]) |
|  | { |
|  | int mid1=(beg+(end-beg))/3; |
|  | int mid2=(end-(end-beg))/3; |
|  |  |
|  | if(ar[mid1]==n) |
|  | { |
|  | return mid1; |
|  | } |
|  | if(ar[mid2]==n) |
|  | { |
|  | return mid2; |
|  | } |
|  | if(n<ar[mid1]) |
|  | { |
|  | end=mid1-1 ; |
|  | return ternary(beg, end, n, ar); |
|  | } |
|  | else if(n>ar[mid2]) |
|  | { |
|  | beg=mid2+1; |
|  | return ternary(beg, end, n, ar); |
|  | } |
|  | else |
|  | { |
|  | beg=mid1+1; |
|  | end=mid2-1; |
|  | return ternary(beg, end, n, ar); |
|  | } |
|  |  |
|  | } |
|  | int main() |
|  | { |
|  | int ar[]={1,2,3,4,5,6,7,8}; |
|  | int beg=0; |
|  | int end=7; |
|  | cout<<"enter number to search"<<endl; |
|  | int n; |
|  | cin>>n; |
|  |  |
|  | cout<<"found at "<<ternary(beg, end, n, ar)<<endl; |
|  | } |