가장 긴 증가하는 부분 수열

1. 문제풀이 방법(알고리즘)

가장 긴 증가 부분 수열의 길이와 그 수열을 구하기 위해서는 수열의 각 원소까지 원소들로 만들어지는 가장 긴 증가 부분 수열의 길이와 각 원소까지 가장 긴 증가 부분 수열이 되도록 하는 바로 전 원소의 위치를 저장하는 동적계획법을 이용해 해결한다. 모든 값을 저장한 이후에는 수열 원소까지의 가장 긴 증가 부분 수열의 길이 중 가장 큰 값(가장 긴 증가 부분 수열의 길이)을 구하고, 그 값의 위치를 이용해 바로 전 위치를 저장한 배열을 이용해서 가장 긴 증가 부분 수열까지 구한다.

1. 소스코드 & 예제
2. **#include <iostream>**
3. **using namespace std;**
4. **#define MAX 1000**
5. **int main(void)**
6. **{**
7. **int n=0;**
8. **int arr[MAX]={0};**
9. **int h[MAX]={0};**
10. **int p[MAX]={0};**
12. **cin>>n;**
14. **for(int i=0; i<n; i++)**
15. **{**
16. **cin>>arr[i];**
17. **}**
18. **for(int i=0; i<n; i++)**
19. **{**
20. **h[i]=1;**
21. **for(int j=0; j<i; j++)**
22. **if(arr[i]>arr[j] && h[i]<=h[j])**
23. **{**
24. **h[i]=h[j]+1;**
25. **p[i]=j;**
26. **}**
27. **}**
29. **int max=h[0];**
30. **int max\_index;**
31. **for(int i=0; i<n; i++)**
32. **{**
33. **if(max<h[i])**
34. **{**
35. **max=h[i];**
36. **max\_index=i;**
37. **}**
38. **}**
39. **cout<<"max height: "<<max<<endl;**
41. **int lis[MAX];**
42. **int index=max;**
43. **int i=max\_index;**
44. **while(index>0)**
45. **{**
46. **lis[--index]=arr[i];**
47. **i=p[i];**
48. **}**
50. **cout<<"max subseq: ";**
51. **for(int i=0; i<max; i++)**
52. **cout<<lis[i]<<" ";**
53. **return 0;**
54. **}**

예제 실행 결과는 다음과 같다.

입력 값:

6

10 20 10 30 20 50

출력 값:

max height: 4

max subseq: 10 20 30 50

1. 소스코드 설명

8~11행에서 선언된 변수는 다음과 같은 의미를 가진다.

정수 n: 입력 받은 수열의 크기

배열 arr: 입력 받은 수열을 저장할 배열

배열 h: h[i]는 arr[i]으로 끝나는 증가 부분 수열 중에 가장 긴 증가 부분 수열의 길이를 저장할 배열

배열 p: p[i]는 arr[i]으로 끝나는 증가 부분 수열 중에 가장 긴 증가 부분 수열에서 arr[i] 바로 전에 나오는 원소의 인덱스를 저장할 배열

n과 arr을 입력 받은 후(13~18행), 배열 h와 p를 구하는 과정(20~29행)을 다음과 같다.

for문을 통해 arr 배열을 돌면서(i←0~n-1), h[i]=1로 해준다. (자기 자신만을 가지는 증가 부분 수열을 무조건 가질 수 있으므로) i 앞에 존재하는 인덱스 중에서 배열의 값이 arr[i]보다 작으면서 h값이 h[i]보다 크거나 같으면 그 값이 해당 인덱스(if문에 해당하는 j)값까지의 증가 부분 수열 중 가장 큰 값이며 그 수열에 arr[i]를 추가한 것이 i까지의 가장 긴 부분 수열이 될 것이므로 h[i]에 해당 h값+1을 저장하는 방식으로 배열 h를 구한다. 이 때 해당 인덱스를 p[i]값에 저장하면 arr[i] 바로 전에 등장하는 원소의 인덱스 값이 저장된다. (동적계획법으로 배열의 값을 저장한다)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| i | 0 | 1 | 2 | 3 | 4 | 5 |
| arr[i] | 10 | 20 | 10 | 30 | 20 | 50 |
| h[i] | 1 | 2 | 1 | 3 | 2 | 4 |
| p[i] | 0 | 0 | 0 | 1 | 0 | 3 |

↑예제를 입력했을 때 배열 값

31, 32행에서는 max, max\_index를 선언하였다.

정수 max: 가장 큰 h[i]값을 구하기 위한 변수

정수 max\_index: 이 때의 i값을 구하기 위한 변수

저장된 배열 h와 새롭게 선언한 변수를 이용해 가장 긴 증가 부분 수열의 길이를 구하는 과정(33~41행)은 다음과 같다.

for문을 이용해서 h[i]의 인덱스를 돌면서 가장 긴 길이를 max에, 그 인덱스 값을 max\_index에 저장한다.

44~46행에서 선언된 변수는 다음과 같은 의미를 가진다.

배열 lis: 최종적으로 구하는 가장 긴 증가 부분 수열을 저장할 배열

정수 index: max-1값부터 시작해서 0이 될 때까지 lis배열의 인덱스를 나타내기 위한 변수(가장 긴 증가 부분 수열의 길이인 max값이 lis배열의 크기이므로)

정수 i: lis배열에 저장하기 위해 max\_ index부터 시작해 그 이전 인덱스 값을 구하기 위한 변수

가장 긴 증가 부분 수열을 구하고 출력하는 과정(47~56행)은 다음과 같다.

index는 max값부터 0이 될 때까지(index←max-1~0) while문을 돌면서, arr[i]값을 저장하는 데 이때 i값은 바로 앞 인덱스의 배열을 저장해둔 p[i]값으로 계속 바꿔주면 lis가 구해진다. 최종적으로 구한 lis배열은 for문을 통해 출력하면 된다.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| index | i | arr[i] | p[i] | lis |
| 4 | 5 | 50 | 3 | lis[3]=50 |
| 3 | 3 | 30 | 1 | lis[2]=30 |
| 2 | 1 | 20 | 0 | lis[1]=20 |
| 1 | 0 | 10 | 0 | lis[0]=10 |

↑예제를 입력했을 때 while문 안에서 lis배열이 저장되는 과정