

Original	Seq2Seq	Tree2Tree
<pre> #define MOD 1000000007 #define ll long long int #define vi vector&lt;int&gt; #define vii vector&lt;vector&lt;int&gt;&gt; #define PI 3.1415926535897932384626433832795 #define INF 9223372036854775807LL  int main() {     ios::sync_with_stdio(false);     cin.tie(NULL);      int n;     cin &gt;&gt; n;     map&lt;int, int&gt; cn;     int ans = 0;      for (int i = 0; i &lt; n; i++) {         int x;         cin &gt;&gt; x;         cn[x]++;         ans = max(ans, cn[x]);     }      cout &lt;&lt; ans;     return 0; } </pre>	<pre> int main() {     int n, m;     cin &gt;&gt; n &gt;&gt; m;     int a[n];      for (int i = 0; i &lt; n; i++) {         cin &gt;&gt; a[i];     }      int ans = 0;     for (int i = 0; i &lt; n; i++) {         if (a[i] &lt;= m) {             ans++;         } else {             break;         }     }      for (int i = n - 1; i &gt;= 0; i--) {         if (a[i] &lt;= m) {             ans++;         } else break;     }      cout &lt;&lt; ans &lt;&lt; endl;     return 0; } </pre>	<pre> int main() {     ios::sync_with_stdio(false);     cin.tie(0);      int n;     cin &gt;&gt; n;     map&lt;int, int&gt; cn = map&lt;int, int&gt;();     int ans = 0;      for (int i = 0; i &lt; n; i++) {         int x;         cin &gt;&gt; x;         cn[x]++;         ans = max(ans, cn[x]);     }      cout &lt;&lt; ans &lt;&lt; endl;     return 0; } </pre>
<pre> #define int long long #define endl '\n' const int maxl = 4e5 + 7;  signed main() {     ios::sync_with_stdio(false);     cin.tie(0);     cout.tie(0);      int n, m;     cin &gt;&gt; n &gt;&gt; m;     vector&lt;int&gt; nana;     for (int i = 0; i &lt; n; i++) {         int x;         cin &gt;&gt; x;         nana.push_back(x);     }      sort(nana.begin(), nana.end());     int ans = 1;     for (int i = 0; i &lt; n; i++) {         int pos = upper_bound(             nana.begin() + i,             nana.end(),             nana[i] + m         ) - nana.begin() - 1;         ans = max(pos - i + 1, ans);     }     cout &lt;&lt; n - ans; } </pre>	<pre> typedef long long ll; typedef long double ld; typedef pair&lt;int, int&gt; pii; typedef pair&lt;ll, ll&gt; pll; typedef vector&lt;int&gt; vi; typedef vector&lt;ll&gt; vll; typedef vector&lt;pii&gt; vpii; typedef vector&lt;pll&gt; vpll; typedef vector&lt;pli&gt; vpli;  int main() {     ios_base::sync_with_stdio(false);     cin.tie();      int n, m;     cin &gt;&gt; n &gt;&gt; m;     vector&lt;int&gt; a(n);      for (int i = 0; i &lt; n; i++) cin &gt;&gt; a[i];      int ans = 0;      for (int i = 0; i &lt; n; i++) {         int j = i;         while (j &lt; n &amp;&amp; a[j] &lt;= a[i] + j) j++;         ans = max(ans, j - i + 1);         i = j;     }      cout &lt;&lt; ans &lt;&lt; endl;     return 0; } </pre>	<pre> const long long maxl = 1e9 + 7;  int main() {     ios::sync_with_stdio(false);     cin.tie(0);     cout.tie(0);      long long n, m;     cin &gt;&gt; n;     vector&lt;long long&gt; nana = vector&lt;long long&gt;();      for (long long i = 0; i &lt; n; i++) {         cin &gt;&gt; nana[i];     }      sort(nana.begin(), nana.end());     long long x = 0;      for (long long ans = 0; ans &lt; n; ans++) {         long long i = 0;         for (long long pos = 0; pos &lt; n; pos++) {             if (ans &amp; (1 &lt;&lt; ans)) {                 i += nana[ans];             }         }         m = max(x, i);     }      cout &lt;&lt; x &lt;&lt; endl;     return 0; } </pre>
<pre> typedef long long ll; const int INF = 1e9; vector&lt;int&gt; arr[200005]; bool vis[100005]; int pay[100005];  int dfs(int u) {     vis[u] = 1;     int ans = pay[u];     for (auto i: arr[u]) {         if (!vis[i]) {             ans = min(ans, dfs(i));         }     }     return ans; }  int main() {     int n, m;     cin &gt;&gt; n &gt;&gt; m;     for (int i = 1; i &lt;= n; i++) cin &gt;&gt; pay[i];     for (int i = 0; i &lt; m; i++) {         int u, v;         cin &gt;&gt; u &gt;&gt; v;         arr[u].push_back(v);         arr[v].push_back(u);     }     ll Ans = 0;     for (int i = 1; i &lt;= n; i++) {         if (!vis[i]) {             Ans += dfs(i);         }     }     cout &lt;&lt; Ans &lt;&lt; endl; } </pre>	<pre> int main() {     int n, m;     cin &gt;&gt; n &gt;&gt; m;     int a[n];      for (int i = 0; i &lt; n; i++) {         cin &gt;&gt; a[i];     }      int ans = 0;     for (int i = 0; i &lt; n; i++) {         if (a[i] &lt;= m) {             ans++;         } else {             break;         }     }      for (int i = n - 1; i &gt;= 0; i--) {         if (a[i] &lt;= m) {             ans++;         } else break;     }      cout &lt;&lt; ans &lt;&lt; endl;     return 0; } </pre>	<pre> const int ll = 1e4 + 5; vector&lt;int&gt; INF[100005]; int arr[100005]; int vis[100005]; int pay[100005];  void dfs(int u, int ans) {     vis[u] = 1;     vis[u] = ans;     for (auto i: vis[u]) {         if (!vis[i]) {             dfs(i, ans);         }     } }  int main() {     int n, m;     cin &gt;&gt; n &gt;&gt; m;     for (int i = 1; i &lt;= n; i++) {         cin &gt;&gt; vis[i];     }     for (int i = 1; i &lt; n; i++) {         int u, v;         cin &gt;&gt; u &gt;&gt; v;         vis[u].push_back(u);         vis[u].push_back(u);     }     long long Ans = 0;     for (int i = 1; i &lt;= n; i++) {         if (!vis[i]) {             pay = 0x7fffffff;             dfs(i);             pay += pay;         }     }     cout &lt;&lt; Ans;     return 0; } </pre>