**MINPRODUCT**

* Nhậnthấyrằngtrongmọitrườnghợp ta chỉcầnquantâmđếnsốlớnnhất, sốlớnnhì, sốbénhấtvàsốbénhìđểtạoratíchnhỏnhấtcóthể
* Đọcvàocácsố, tìmsốlớnnhất, sốlớnnhì, sốbénhất, sốbénhì, xétcáctrườnghợpcủadấucủacácsốđểđểtìmrakếtquả.

**SHOPS**

Xétbàitoánphụ: chodãysố a có n phầntử, tìmđoạn con dàinhấtcủa a saochôtổngcủađoạnnàyphảilớnhơnhoặcbằngmộtgiátrị p chotrước

* Ta sẽtạomảngtổng Si = a1 + a2 + . . . + aivà S0=0. Nhưvậycôngviệccủachúng ta làsẽtìmcặpchỉsố (i , j) thỏamãncácđiềukiệnsau:
  + 0 <i< j < n.
  + Sj – Si > P.
  + ( j – i )max .
* Trênmảng S, ta sẽtiếnhànhtìmmộtdãygiảmdầnbắtđầutừ S0 vàđánhdấunó, thựchiệnnónhưsau:

memset( ok, false, sizeof(ok));

ok[0]=true; minn=0;

for (inti=1; i<=n; i++)

if (s[i]<minn)

{

ok[i]=true; minn=s[i];

}

* Bâygiờ, chúng ta mớitiếnhànhtrượt, ở đây ta sẽtrượtlùi,do đó ta sẽgánvtd=n. Vịtrícuốithì ta sẽchọnmộtvítríigầnnhấtthỏa ok[i]=true kèmtheođiềukiệni<tvd.
* Bởivìnếukhôngchọnvịtrícó ok[i]= true gầnthìtứclà ta phảichọnmộtvịtrínằmtrongkhoảng [vtd…i]. Lúcđóchắcchắnmộtvịtrí j bấtkìđượcchọnsẽcóSj> Si vìnếuSj< Si thìchắcchắnlúctìmdãy con giảmdàinhấtbắtđầutừ S0 thìphảibổ sung Sjvàsau Si vàgầnvtdhơn.
* Tiếptục, vìSj> Si chonênSvtd – Sj<Svtd - Si màlạicó n – j < n – i.Cảgiátrịvàđộdàicủađoạn [vtd…j] đềukémtốiưuhơnđoạn [vtd…i].
* Do đó, điềuđầutiên ta phảichọnmộtvịtríigầnvtdnhấtcó ok[i] = true. Sauđó ta đem so sánhgiátrịSvtd – Si với P. Sẽcóhaitrườnghợpxảyra:
  + NếuSvtd – Si > P thì ta sẽcậpnhậtlạikq=max(kq , vtd-i) vàtiếptụcgiảmiđểtìmvịtrí j tốiưunhấtvài> j. Dĩnhiên, ta nghĩđếnngayđósẽlàvịtrícó ok[j]=true. Lùivềvịtrí j chonên [vtd…j] chắcchắnsẽdàihơn [vtd … i] nhưngSvtd – Sj<Svtd – Si nênnếumàSvtd – Sj> P thì ta tiếptụccậpnhậtvàlùicònkhôngthìdừnglại. Tấtnhiênđiềukiệnlùivềsẽkếtthúc ở S0.
  + NếuSvtd – Si < P khiđó, nếu ta cứgiảmixuốngthìvịtrítốtưunhấtvẫnnhư ở trườnghợpmộtlà j để ok[j]=true, nhưvậythìgiátrịSvtd – Sj<Svtd – Si < P. Do đólúcnày ta sẽlùivtd. Cứnhưvậyđểkếtthúcbàitoán !

**EDGES**

* Nhậnthấyrằng ta lấycàngnhiềucạnhâmcàngtốtvìcànglấythêmnhiềucạnhâmthì chi phícàngnhỏvàsốcạnhcủađồthịcàngtăng.
* Luônluônlấytấtcảcạnhâm.
* Saukhilấytấtcảcạnhâm, ta chuyểnđồthịvềdạng DAG, mỗiđỉnhcủađồthịmớilàmột TPLT mạnhtrênđồthịcũ.
* Nếusốđỉnhcủađồthịmớitạobằng 1, tứclàđồthị ban đầulàliênthôngmạnhthì ta in rakếtquảbằngtổngtrọngsốcáccạnhâm, kếtthúcchươngtrình.
* TÌm 2 đỉnh S và T trênđồthị DAG vừatạo. ( sốcạnhvàocủa DAG[S] bằng 0, sốcạnhracủa DAG[T] bằng 0 )
* Phầncònlạichỉlàtìmđộdàiđườngđingắnnhấttừđỉnh T đếnđỉnh S, kếtquảđưarasẽbằngtổngtrọngsốcủacáccạnhâmcộngvớiđộdàicủađườngđingắnnhấttừđỉnh T đếnđỉnh S.

**Dance**

Đọc code đoạn này nhé:  
<https://pastebin.com/uzCs8QCS>

Ta sẽ cứ lấy 3 thằng đầu, tạo thêm 1 thằng ở cuối, nối thằng cuối -> 3 thằng đầu.

Do n lẻ nên là sẽ tạo đến khi còn 1 thằng.

Bây h ta có 1 cái cây ( surprise mofk). Thằng nhảy với công chúa sẽ là nút 1, các lá sẽ là n thằng ban đầu.

Chặt nhị phân kết quả ( gọi là mid) . Các số >= mid -> 1, < mid 0.

Ta cần chọn sao cho giá trị của nút 1 là 1.Do mỗi nút ( trừ lá) có 3 con, nên là ta cần làm sao để có ít nhất 2 con của nó có giá trị là 1.

Đến đây thì ta làm DP trên cây như sau :

Dp[u] = số lượng số 1 min để u có giá trị là 1.

Công thức:

Nếu u khác lá:

Dp[u] = min(dp[vi] + dp[vj]) ( vi và vj là 2 con của u)

Else

* Nếu u đã có giá trị ( gán từ ban đầu) dp[u] = (val[u] == 1 ? 0 : inf).
* Else, dp[u] = 1.

Kết quả của lần chặt là đúng nếu dp[1] <= số thằng 1.