Misket Çözüm

Altgörev 1 (12 puan)

Baha Dede m misketi t toruna dağıtırsa üzülen torun sayısı $uzulen(m,t)$ olsun. Eğer m, t'nin tam katıysa $uzulen(m,t) = 0$ olur. Değilse diğerlerinden fazla misket alacak kişi sayısı $m\%t$ olacaktır. Eğer bu sayı c'den küçükse Baha Dede cebine koyup kimseyi üzmeyebilir. Değilse üzülecek kişi sayısı $t - m\%t$ olacaktır. $m < t$ ise cebine koymayacaktır. Bu şekilde bütün $m_l \leq m \leq m_r$ ve $t_l \leq t \leq t_r$ için $uzulen(m,t)$ hesaplanır ve en uygun m seçilir.

Altgörev 2 (11 puan)

$uzulen(m,t)$ hesaplarken eğer $t \leq c$ ise her zaman $m\%t \leq c$ olacağı için $uzulen(m,t) = 0$ olur. Bu $m < t$ için geçerli değildir. Ancak bu durumda da en büyük t her zaman daha kötü cevap getireceği için sadece $t > c$ için cevapları hesaplarsak her m için 10 kere hesaplamış oluruz.

Altgörev 3 (15 puan)

$5 * 10^5$'ten küçük t'ler için 1'den başlarak artan m'ler üzülen torun sayısını bir zaman için azaltır, daha sonra 0'da sabit kalır. Çünkü t'den sonra alınan misketler için cebe koyarak bir önceki t'nin katına indirilebilir. Bu sebeple en son m'yi seçmek her zaman doğru sonucu verecektir.
$5 * 10^5$'ten büyük t değerleri için de aynısı geçerlidir. Çünkü 1'den t'ye kadar olan m değerleri için üzülen torun sayısı azalarak gider ve t'den büyük olan m değerlerini de cebe koyarak t'ye indirilebilir. Çünkü $10^6 - t$ her zaman $5 * 10^5$'ten küçüktür. Bu sebeple yine en son m'yi almak en mantıklısıdır. Sonuç olarak m_r değeri doğru cevaptır.

Altgörev 4 (22 puan)

Bir m için bütün t'lerin üzülen torun sayısını sırayla yazarsak şöyle olur:
$0, 0, 0, ..., 0, t-c-1, t-c-2, t-c-3, ..., 2, 1, 0, 0, ..., 0, t-c-1, t-c-2, ...$
Amacımız en kötü ihtimalin 0 olduğu m'lere bakmak olduğu için 0 olmayan değerleri tek seferde atlayabiliriz. Bunun için $m\%t = c+1$ olan t'lerden $m\%t = t-1$ olanları tek seferde 1 yaparız. Bir m için m/t adet işlem yaparız.

Altgörev 5 (15 puan)

Örnek olarak 10 12 80 100 0 girdisinin $m - t$ grafiğini yazalım

0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1
3	2	1	0	10	9	8	7	6	5	4	3	2	1	0	10	9	8	7	6
4	3	2	1	0	11	10	9	8	7	6	5	4	3	2	1	0	11	10	9

0. sütuna 0, 1. sütuna 1, 2. sütuna 2... eklersek:

0	10	10	10	10	10	10	10	10	10	10	20	20	20	20	20	20	20	20	20
3	3	3	3	14	14	14	14	14	14	14	14	14	14	25	25	25	25	25	25
4	4	4	4	4	16	16	16	16	16	16	16	16	16	16	16	28	28	28	28

Her sütunun maksimumunu bulmak son tabloda daha kolay, bulduğumuz sonuçtan index sayısını çıkarırsak (başta eklemiştik) cevabı elde ederiz. Bunun nasıl bulunacağı tam çözümde anlatılmıştır.

Tam Çözüm (25 puan)

Yine örnek olarak 10 12 80 100 5 girdisinin $m - t$ grafiğini yazalım, ancak saklanan misket sayılarını da eksi olarak alacağız (3, 2, 1, 0'dan sonra gelen 0'lar için eksi değerlere ineceğiz).

0	-1	-2	-3	-4	-5	4	3	2	1	0	-1	-2	-3	-4	-5	4	3	2	1
-3	-4	-5	5	4	3	2	1	0	-1	-2	-3	-4	-5	5	4	3	2	1	0
4	3	2	1	0	-1	-2	-3	-4	-5	6	5	4	3	2	1	0	-1	-2	-3

Yine sütunlara index sayılarını eklersek:

0	0	0	0	0	0	10	10	10	10	10	10	10	10	10	10	20	20	20	20
-3	-3	-3	8	8	8	8	8	8	8	8	8	8	8	19	19	19	19	19	19
4	4	4	4	4	4	4	4	4	4	16	16	16	16	16	16	16	16	16	16

İlk sütundaki değerleri hesaplayacağız. Sonraki sütundakilerin hepsini hesaplamamıza gerek yok çünkü çoğu zaman sayılar değişmiyor. Sadece bir sayıya güncelleme geldiği zaman hesaplamamız yetiyor. Güncelleme gelen sayılar, fazladan misket alan torun sayısının $c + 1$ olduğu zamana denk geliyor. Yani $m\%t = c + 1$ olduğu zamana. t'yi yalnız bırakırsak
$m = c + 1 + t * k$
$m - c - 1 = t * k$
Yani her m için $m - c - 1$'i bölen t değerlerinde güncelleme olacak. Ve güncellenen sayılar $t - c - 1 + index$ olacak. Amacımız bunların en büyüğünü bulmak olduğu için sadece en büyük t'yi hesaplamak yeterli olacaktır. Ancak her m için t_l ve t_r arasındaki en büyük böleni hesaplamak çok yavaş olduğu için bunu başta sieve ile yapmamız gerekiyor.