Posiadłość (rozwiązanie)

XIV OIJ, próbne zawody II stopnia
29 lutego 2020

Kod zadania: **pos**
Limit czasu: **15 s**
Limit pamięci: **256 MB**

Olimpiada Informatyczna Juniorów

Zadanie pochodzi z Wrocławskich Sparingów Informatycznych (`https://solve.edu.pl/~sparingi/`). Omówienie video znajduje się pod adresem: `https://www.youtube.com/watch?v=sp8n_5k2Ry8`. Gorąco zachęcamy do udziału w sparingach, które stanowią świetne przygotowanie do Olimpiady Informatycznej Juniorów.

W zadaniu, dla podanego na wejściu przedziału liczb naturalnych od A do B włącznie, należało znaleźć liczbę x, w tym przedziale, której suma cyfr jest jak największa możliwa (tak naprawdę, wystarczy nam wiedzieć jaka jest suma cyfr liczby x, a niekoniecznie musimy znać samą tę liczbę).

Rozwiązanie za 22% punktów

W pierwszym podzadaniu, różnica między wartościami A oraz B nie przekracza miliona. To znaczy, że w przedziale jest co najwyżej $1\,000\,001$ liczb i można sobie pozwolić na sprawdzenie sumy cyfr dla każdej z nich.

Obliczenie sumy cyfr można wykonać przez systematyczne dzielenie liczby przez 10 i dodawanie do wyniku ostatniej jej cyfry (którą można uzyskać obliczając resztę z dzielenia liczby przez 10).

pos1.py

```python
def suma_cyfr(x):
  if x == 0: return 0
  return (x % 10) + suma_cyfr(x // 10)


a, b = map(int, input().split())
print(max([suma_cyfr(x) for x in range(a, b+1)]))
```

pos1.cpp

```cpp
#include <bits/stdc++.h>
using namespace std;

int suma_cyfr(long long x) {
  if (x == 0) return 0;
  return (x % 10) + suma_cyfr(x / 10);
}

int main() {
  ios_base::sync_with_stdio(false);
  cin.tie(0);

  long long a, b;
  cin >> a >> b;
  int najlepszy_wynik = 0;
  for (long long j = a; j <= b; j++)
    najlepszy_wynik = max(najlepszy_wynik, suma_cyfr(j));
  cout << najlepszy_wynik << "\n";

  return 0;
}
```

W języku C++ należy pamiętać o użyciu typu `long long`.

Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".

Rozwiązanie za 44% punktów

W drugim podzadaniu $A = 1$, czyli zadanie sprowadza się po prostu do znalezienia liczby x nie przekraczającej B o największej sumie cyfr (nie musimy się więc przejmować, że uzyskana liczba będzie mniejsza niż A).

Rozważmy liczbę B: niech jej najbardziej znaczącą cyfrą jest $c \geq 1$. Jeśli zmniejszymy ją o 1, to wszystkie cyfry po niej będziemy mogli ustawić na dziewiątki. Nie może być więc możliwe uzyskanie istotnie lepszej (większej o więcej niż 1) sumy cyfr niż ten sposób.

Inne opcje to pozostawić pierwszą cyfrę bez zmian i manipulować kolejnymi cyframi. Ponownie jednak, piewszą zmienianą cyfrę musimy zmniejszyć i dopiero wtedy możemy wszystkie mniej znaczące cyfry ustawić na 9 (co nie da lepszego wyniku niż poprzedni pomysł).

Dochodzimy to wniosku, że albo pomysł zmniejszenia pierwszej cyfry był dobry albo nie należy zmieniać żadnej cyfry i ustalić $x = B$ (tak dzieje się wtedy, gdy liczba B już kończy się samymi dziewiątkami, wtedy zmniejszenie najbardziej znaczącej cyfry nic nie daje i niepotrzebnie zmniejsza sumę cyfr).

pos2.py

```python
def suma_cyfr(x):
  if x == 0: return 0
  return (x % 10) + suma_cyfr(x // 10)


def liczba_cyfr(x):
  if x == 0: return 0
  return 1 + liczba_cyfr(x // 10)


def cyfra_znaczaca(x):
  if x >= 10: return cyfra_znaczaca(x // 10)
  return x


a, b = map(int, input().split())
wynik1 = suma_cyfr(b)
wynik2 = (cyfra_znaczaca(b) - 1) + 9 * (liczba_cyfr(b) - 1)
print(max(wynik1, wynik2))
```

pos2.cpp

```cpp
#include <bits/stdc++.h>
using namespace std;

int suma_cyfr(long long x) {
  if (x == 0) return 0;
  return (x % 10) + suma_cyfr(x / 10);
}

int liczba_cyfr(long long x) {
  if (x == 0) return 0;
  return 1 + liczba_cyfr(x / 10);
}

int cyfra_znaczaca(long long x) {
  if (x >= 10) return cyfra_znaczaca(x / 10);
  return x;
}

int main() {
  ios_base::sync_with_stdio(false);
  cin.tie(0);

```


Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".

```
  long long a, b;
  cin >> a >> b;  // program i tak jest w wersji dla a = 1
  int wynik1 = suma_cyfr(b);
  int wynik2 = (cyfra_znaczaca(b) - 1) + 9 * (liczba_cyfr(b) - 1);
  cout << max(wynik1, wynik2) << "\n";

  return 0;
}
```

Rozwiązanie wzorcowe

Liczbę A w razie potrzeby dopełniamy zerami z lewej strony, żeby miała taką samą długość jak liczba B.

Kontynuujemy obserwacje z rozwiązania powyżej. Ponownie, jednym z kandydatów na optymalną wartość liczby x jest liczba B. Natomiast w przypadku, w którym $x < B$ musimy zadbać o to, żeby znaleziony x nie był mniejszy niż A. Nie zawsze więc możemy pozwolić sobie na zmniejszenie najbardziej znaczącej cyfry B o 1 (dzieje się tak wtedy, gdy odpowiadająca cyfra w liczbie A jest taka sama). Tak długo więc jak cyfry liczby A i B są równe, nie możemy nic zrobić. Chcąc jednak uzyskać jak najwięcej dziewiątek, cyfra, którą chcemy zmniejszyć to pierwsza pozycja, w której odpowiadające cyfry w liczbach A i B są różne.

pos3.py

```
def cyfry_liczby(x):
  cyfry = []
  for _ in range(19):
    cyfry.append(x % 10)
    x //= 10
  cyfry = cyfry[::-1]
  return cyfry


def popraw(cyfry_a, cyfry_b):
  cyfry_c = list(cyfry_b)
  juz_dziewiatki = False
  for i in range(19):
    if juz_dziewiatki:
      cyfry_c[i] = 9
    elif cyfry_a[i] != cyfry_b[i]:
      cyfry_c[i] = cyfry_b[i] - 1
      juz_dziewiatki = True
  return cyfry_c


a, b = map(int, input().split())
cyfry_a, cyfry_b = cyfry_liczby(a), cyfry_liczby(b)
cyfry_c = popraw(cyfry_a, cyfry_b)
wynik1 = sum(cyfry_b)
wynik2 = sum(cyfry_c)
print(max(wynik1, wynik2))
```

pos3.cpp

```
#include <bits/stdc++.h>
using namespace std;

int suma_liczb(const vector<int>& liczby) {
  int suma = 0;
  for (int x : liczby)
    suma += x;
  return suma;
```


Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".

```cpp
}

vector<int> cyfry_liczby(long long x) {
  vector<int> cyfry;
  for (int i = 0; i < 19; i++) {
    cyfry.push_back(x % 10);
    x /= 10;
  }
  reverse(cyfry.begin(), cyfry.end());
  return cyfry;
}

vector<int> popraw(const vector<int>& cyfry_a, const vector<int>& cyfry_b) {
  vector<int> cyfry_c = cyfry_b;
  bool juz_dziewiatki = false;
  for (int i = 0; i < 19; i++) {
    if (juz_dziewiatki)
      cyfry_c[i] = 9;
    else if (cyfry_a[i] != cyfry_b[i]) {
      cyfry_c[i] = cyfry_b[i] - 1;
      juz_dziewiatki = true;
    }
  }
  return cyfry_c;
}

int main() {
  ios_base::sync_with_stdio(false);
  cin.tie(0);

  long long a, b;
  cin >> a >> b;
  vector<int> cyfry_a = cyfry_liczby(a), cyfry_b = cyfry_liczby(b);
  vector<int> cyfry_c = popraw(cyfry_a, cyfry_b);
  int wynik1 = suma_liczb(cyfry_b);
  int wynik2 = suma_liczb(cyfry_c);
  cout << max(wynik1, wynik2) << "\n";

  return 0;
}
```


Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".