**Wojciech Rytter** Treść zadania, Opracowanie
**Piotr Smulewicz** Program

**Dostępna pamięć: 256 MB.** *OI, etap III, dzień drugi, 14.04.2016*

# Klubowicze

*Bajtocki Klub Dyskusyjny jest wyjątkowy pod każdym względem. Posiada on $2^n$ członków, z których każdy zadeklarował, jakie ma poglądy na $n$ fundamentalnych pytań. Konkretne sformułowanie pytań nie jest istotne, wystarczy wiedzieć, że są to pytania, na które można udzielić jednej z dwóch odpowiedzi (np. „kawa czy herbata?”). Poglądy danej osoby możemy kodować za pomocą ciągu bitów, który interpretowany w systemie binarnym da liczbę całkowitą z przedziału od $0$ do $2^n - 1$.*

*W klubie nie ma dwóch osób o jednakowych poglądach. Powiemy, że dwie osoby są* **prawie zgodne**, *jeśli ich poglądy różnią się tylko na jednym pytaniu. Ponadto klubowicze to $2^{n-1}$ panów i $2^{n-1}$ pań, którzy tworzą $2^{n-1}$ par. Klubowicze spotykają się przy* **okrągłym** *stole. Chcemy ich tak usadzić, żeby każdy klubowicz siedział obok swojej partnerki lub swojego partnera, a obok siebie po drugiej stronie miał osobę prawie zgodną.*

## Wejście

*W pierwszym wierszu standardowego wejścia znajduje się liczba całkowita $n$ oznaczająca liczbę fundamentalnych pytań. W kolejnych $2^{n-1}$ wierszach znajdują się opisy par klubowiczów: w $i$-tym z tych wierszy znajdują się dwie liczby całkowite $a_i$, $b_i$ ($0 \leqslant a_i, b_i \leqslant 2^n - 1$) oddzielone pojedynczym odstępem, oznaczające, że klubowicze o zestawie poglądów opisanym liczbami $a_i$ i $b_i$ są parą. Każda liczba reprezentująca klubowicza pojawi się na wejściu dokładnie raz.*

## Wyjście

*Jeśli nie istnieje usadzenie klubowiczów spełniające warunki zadania, to w jedynym wierszu standardowego wyjścia należy wypisać jedno słowo* `NIE`.

*Jeśli takie usadzenie istnieje, to w jedynym wierszu standardowego wyjścia należy wypisać ciąg $2^n$ liczb całkowitych pooddzielanych pojedynczymi odstępami, oznaczający poprawne usadzenie klubowiczów przy okrągłym stole.*

*Jeśli istnieje wiele poprawnych odpowiedzi, należy wypisać dowolną z nich.*

## Przykład

*Dla danych wejściowych:*

```
3
0 5
4 1
3 6
7 2
```

*poprawnym wynikiem jest:*

```
0 5 7 2 6 3 1 4
```

**Testy „ocen”:**

`1ocen`: *$n = 4$, jeśli $i$ jest parzyste, to klubowicze o numerach $i$ oraz $i + 1$ są w parze;*

`2ocen`: *$n = 10$, jeśli $i$ jest nieparzyste, to klubowicze o numerach $i$ oraz $i + 1$ są w parze; wyjątkiem jest klubowicz $2^n - 1$, który jest w parze z klubowiczem $0$;*

`3ocen`: *$n = 15$, test losowy, pary na wejściu są posortowane rosnąco względem liczb $a_i$.*

## Ocenianie

*Zestaw testów dzieli się na 18 grup, z których każda warta jest 5 albo 6 punktów. W grupie numer $k$ znajdują się wyłącznie testy z $n = k + 1$ (a zatem $2 \leqslant n \leqslant 19$).*

# Rozwiązanie

Opiszemy rozwiązanie w terminach grafów nieskierowanych, cykli Hamiltona i skojarzeń doskonałych. Dla przypomnienia, cykl Hamiltona to cykl przechodzący przez każdy wierzchołek grafu dokładnie raz. Z kolei skojarzenie w grafie to podzbiór krawędzi, w którym żadne dwie krawędzie nie mają wspólnego końca. Skojarzenie nazywamy *doskonałym*, gdy każdy wierzchołek grafu jest skojarzony (tzn. jest końcem pewnej krawędzi ze skojarzenia).

Rozważmy graf $\mathcal{H}_n$, którego wierzchołki stanowią liczby $0 \leqslant i < 2^n$ traktowane jako binarne ciągi $n$-elementowe, zaś krawędzie łączą pary liczb różniące się na jednym bicie. Taki graf nazywany jest często *hiperkostką* $n$-wymiarową jako uogólnienie kwadratu i sześcianu na wyższe wymiary. Natomiast niech $\mathcal{K}_n$ będzie *grafem pełnym* o tym samym zbiorze wierzchołków co $\mathcal{H}_n$, w którym każde dwa różne wierzchołki są połączone krawędzią.

Klubowicze z naszego zadania odpowiadają wierzchołkom grafów. W zadaniu mamy dany zbiór par wierzchołków (skojarzenie doskonałe grafu $\mathcal{K}_n$) i chcemy znaleźć cykl taki, że wierzchołki z każdej pary będą leżały obok siebie na cyklu oraz jeśli wierzchołki leżące obok siebie na cyklu nie są jedną z wejściowych par, to ich numery różnią się dokładnie na jednym bicie (skojarzenie doskonałe grafu $\mathcal{H}_n$).

W terminach grafowych oryginalny problem jest równoważny następującemu:

**Wejście:** skojarzenie doskonałe $X$ w grafie $\mathcal{K}_n$

**Wyjście:** cykl Hamiltona $C = X \cup Y$ w grafie $\mathcal{K}_n$ taki, że $Y$ jest skojarzeniem w $\mathcal{H}_n$

Inaczej mówiąc, mając doskonałe skojarzenie w grafie pełnym, chcemy je dopełnić krawędziami z hiperkostki do cyklu Hamiltona. O ile dla grafów niebędących hiperkostką może to być niewykonalne, to w trakcie konstruowania algorytmu przekonamy się, że dzięki specyficznej strukturze grafu $\mathcal{H}_n$ zawsze jesteśmy w stanie znaleźć rozwiązanie.

Głównym pomysłem jest rekurencyjne sprowadzenie problemu do obliczeń na dwóch podhiperkostkach $\mathcal{H}_{n-1}$. Na rozwiązanie składać się będzie: sprytne dołożenie pomocniczych krawędzi, obliczenie rekurencyjnie cykli w podhiperkostkach, następnie usunięcie pomocniczych krawędzi i dołożenie krawędzi ze zbioru krawędzi łączących podhiperkostki.

Podobnie jak wierzchołki sześcianu można podzielić na dwie grupy znajdujące się na przeciwległych ścianach, tak hiperkostkę $\mathcal{H}_n$ można podzielić na dwa podgrafy izomorficzne z $\mathcal{H}_{n-1}$. W tym celu wystarczy ustalić dowolny indeks $i$ oraz pogrupować ciągi bitowe odpowiadające wierzchołkom $\mathcal{H}_n$ w zależności od wartości $i$-tego bitu.

Pokażemy, że dla każdych danych wejściowych istnieje rozwiązanie. Jak się nierzadko okazuje, od dowodu niedaleka droga do algorytmu. Zaczniemy od jednej przydatnej definicji. Jeśli $C$ jest cyklem oraz $Z$ jest podzbiorem krawędzi $C$ i jednocześnie skojarzeniem, to przez $Seq(C, Z)$ oznaczmy listę krawędzi $Z$ w kolejności i skierowaniu zadanym przez $C$ (wybieramy dowolne skierowanie cyklu). Dla przykładu jeśli $C = (1, 2, 4, 3, 6, 5, 1)$ oraz $Z = [(5, 6),\ (2, 4)]$, to $Seq(C, Z) = [(2, 4),\ (6, 5)]$.

**Twierdzenie 1.** *Każde skojarzenie doskonałe w grafie $\mathcal{K}_n$ można dopełnić do cyklu Hamiltona skojarzeniem z grafu $\mathcal{H}_n$.*

**Dowód:** Indukcja po $n$. Za bazę indukcyjną potraktujemy przypadek dwuwymiarowy. Wtedy graf $\mathcal{H}_n$ jest kwadratem, dla którego teza jest oczywista.

Przypuśćmy, że dla $n \geqslant 3$ dokonaliśmy podziału hiperkostki $\mathcal{H}_n$ na dwie podhiperkostki $H_1$ i $H_2$, obie izomorficzne z $\mathcal{H}_{n-1}$. Niech $X$ będzie doskonałym skojarzeniem grafu $\mathcal{K}_n$ i podzielmy krawędzie tego skojarzenia na trzy zbiory:

$$X_1 = X \cap H_1, \qquad X_2 = X \cap H_2, \qquad Z = X - X_1 - X_2.$$

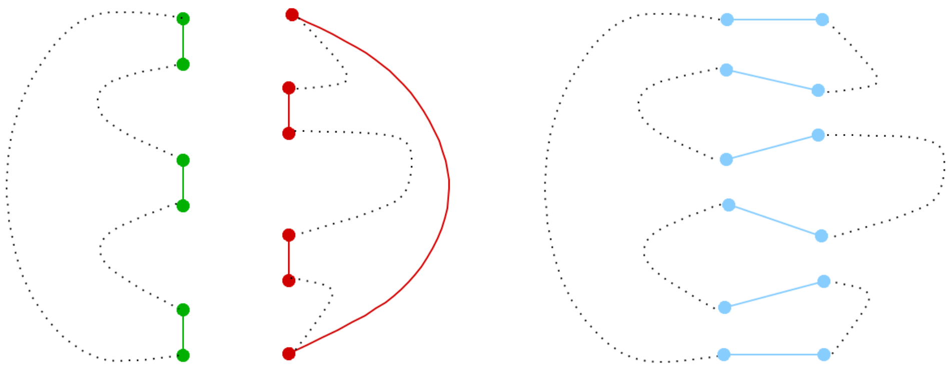
Licząc końce krawędzi w $H_1$ i $H_2$, dostajemy $2 \cdot |X_1| + |Z| = 2 \cdot |X_2| + |Z| = 2^{n-1}$, skąd wnioskujemy, że zbiór $Z$ musi zawierać parzyście wiele krawędzi. Są to niejako łączniki pomiędzy podhiperkostkami, które zostaną użyte do połączenia dwóch cykli, które rekurencyjnie skonstruujemy w podhiperkostkach.

Niech dla $i = 1, 2$ zbiór $D_i$ zawiera końce krawędzi ze zbioru $Z$, które leżą w podhiperkostce $H_i$. Połączmy elementy $D_i$ w dowolny sposób w pary, uzyskując skojarzenie $M_i$. Wtedy $X_i \cup M_i$ jest doskonałym skojarzeniem w grafie izomorficznym z $\mathcal{K}_{n-1}$, zatem z założenia indukcyjnego wynika, że uda nam się znaleźć skojarzenie $Y_i$ w $H_i$, takie że $C_i = X_i \cup M_i \cup Y_i$ będzie cyklem.

Otrzymamy tym samym dwa rozłączne cykle $C_1$ i $C_2$ w $\mathcal{H}_n$, które chcielibyśmy połączyć przy pomocy krawędzi z $Z$. Aby jednak to było możliwe, musimy nieco uważniej skonstruować skojarzenia $M_i$. Niech $\phi\colon D_1 \to D_2$ będzie przyporządkowaniem końcom krawędzi w $Z$ leżących w $H_1$ ich końców leżących w $H_2$. Skojarzenie $M_1$ możemy wybrać dowolnie, ale skojarzenie $M_2$ będzie wyznaczone na podstawie cyklu $C_1$. A konkretnie: dla $Seq(C_1, M_1) = [(v_1, v_2), (v_3, v_4), \ldots, (v_{2k-1}, v_{2k})]$ budujemy cykl $C_2 = X_2 \cup M_2 \cup Y_2$, wybierając

$$M_2 = [(\phi(v_2), \phi(v_3)), (\phi(v_4), \phi(v_5)), \ldots, (\phi(v_{2k}), \phi(v_1))].$$

Teraz możemy już połączyć cykle $C_1$ i $C_2$, zastępując zbiór pomocniczych krawędzi $M_1 \cup M_2$ przez krawędzie zbioru $Z$. Dzięki temu uzyskujemy zbiór $C = X_1 \cup Y_1 \cup X_2 \cup Y_2 \cup Z$. Aby pokazać, że jest on cyklem, wystarczy zauważyć, że powstaje on z $C_2$ poprzez zastąpienie każdej krawędzi $(\phi(v_i), \phi(v_{i+1})) \in M_2$ przez ścieżkę $\phi(v_i) \to v_i \rightsquigarrow v_{i+1} \to \phi(v_{i+1})$, leżącą poza $H_2$. Każda taka zamiana jest



Rys. 1: Ilustracja połączenia cykli $C_1$ i $C_2$. Po lewej stronie linie ciągłe oznaczają skojarzenia $M_1$ i $M_2$. Przerywane linie oznaczają ścieżki leżące wewnątrz podhiperkostek $H_1$, $H_2$. Po prawej stronie skojarzenie $M_1 \cup M_2$ zostało zamienione na $Z$.

niezależna od pozostałych i żadna nie narusza spójności cyklu. Niniejsza konstrukcja została zobrazowana na rysunku 1.

W powyższym rozumowaniu łatwo przeoczyć jeden detal. Przyjęliśmy mianowicie ciche założenie, że zbiór $Z$ jest niepusty. Jest ono istotne, bo w przeciwnym wypadku nie udałoby się nam połączyć obu cykli. Możemy jednak łatwo skonstruować taki podział $\mathcal{H}_n$ na dwie podhiperkostki, w którym to założenie jest spełnione. Wystarczy wybrać dowolną krawędź $e \in X$ i znaleźć bit, na którym jej końce się różnią, a następnie podzielić $\mathcal{H}_n$ w zależności od wartości tego bitu. W ten sposób zagwarantujemy, że co najmniej krawędź $e$ będzie należeć do zbioru $Z$. ■

Dowód twierdzenia łatwo przetłumaczyć na algorytm rekurencyjny. Poniżej przedstawiamy pseudokod funkcji przyjmującej jako argumenty $n$, hiperkostkę wymiaru $n$ oraz skojarzenie $X$, i zwracającej szukany cykl Hamiltona.

**function** FindCycle$(n, H, X)$<br>
**begin**<br>
  **if** $n = 2$ **then**<br>
    **return** sprawdź oba skojarzenia doskonałe $H$ i wybierz to pasujące do $X$;<br>
  $(H_1, H_2) :=$ podział $H$ na podhiperkostki z niepustym łącznikiem;<br>
  $X_1 := X \cap H_1,\ X_2 := X \cap H_2,\ Z := X - X_1 - X_2$;<br>
  $D_1 :=$ zbiór wierzchołków w $H_1$ będących końcami krawędzi z $Z$;<br>
  $M_1 :=$ dowolne skojarzenie doskonałe w grafie pełnym na zbiorze $D_1$;<br>
  $C_1 :=$ FindCycle$(n-1,\ H_1,\ X_1 \cup M_1)$;<br>
  $[(v_1, v_2), (v_3, v_4), \ldots, (v_{2k-1}, v_{2k})] := Seq(C_1, M_1)$;<br>
  $M_2 := [(\phi(v_2), \phi(v_3)), (\phi(v_4), \phi(v_5)), \ldots, (\phi(v_{2k}), \phi(v_1))]$;<br>
  $C_2 :=$ FindCycle$(n-1,\ H_2,\ X_2 \cup M_2)$;<br>
  **return** cykl Hamiltona złożony z $(C_1 - M_1)\ \cup\ (C_2 - M_2)\ \cup\ Z$;<br>
**end**

Dla ustalonego $n$ wszystkie zbiory występujące w funkcji FindCycle mają co najwyżej $N = 2^{n-1}$ elementów, a potrzebne operacje możemy wykonać w czasie liniowym od ich rozmiaru. Złożoność obliczeniową algorytmu można zatem opisać równaniem

$$T(N) = 2 \cdot T(N/2) + O(N),$$

które jest spełnione dla $T(N) = O(N \log N)$. Z rekurencją tego typu można się spotkać na przykład w klasycznym problemie sortowania przez scalanie (ang. *mergesort*). Jest to szczególny przypadek *Twierdzenia o rekurencji uniwersalnej* [6]. Tak więc ostatecznie złożoność rozwiązania to $O(2^{n-1} \log 2^{n-1}) = O(n2^n)$.

Istotną częścią implementacji jest interpretacja rozkładu hiperkostki w języku operacji na maskach bitowych. Kod z komentarzami objaśniającymi poszczególne operacje można znaleźć w pliku `klu2.cpp`.