Sztuczka meet-in-the-middle i logarytm dyskretny

Bartosz Chomiński

1 Wstęp

Klasyczny problem wydawania reszty to problem, w którym mając n nominałów monet o wartościach $a_1, a_2, \ldots, a_n$ i kwotę k, mamy za zadanie wyznaczyć najmniejszy podzbiór posiadanych monet, którego wartości monet sumują się dokładnie do k.

Rozwiążemy problem wydawania reszty w złożoności czasowej $\mathcal{O}(n \cdot 2^{n/2})$, ale kosztem zwiększonej złożoności pamięciowej rzędu $\mathcal{O}(2^{n/2})$, gdzie n jest liczbą posiadanych nominałów.

2 Przyspieszenie w problemie wydawania reszty

2.1 Algorytm

Niech $a_1, a_2, \ldots, a_n$ będą nominałami monet w instancji problemu wydawania reszty i niech k będzie kwotą do wydania. Podzielmy ten worek monet na dwie w miarę równe części i rozdajmy rodzeństwu: Jasiowi i Małgosi. Monety $a_1, a_2, \ldots, a_{\lceil n/2 \rceil}$ otrzyma Jaś, a monety $a_{\lceil n/2 \rceil + 1}, a_{\lceil n/2 \rceil + 2}, \ldots, a_n$ otrzyma Małgosia.

Najpierw Jaś wyznaczy i zapamięta wszystkie kwoty jakie jest w stanie wydać posiadanymi przez siebie monetami oraz w ile co najmniej nominałów jest w stanie wydać każdą kwotę.

Następnie Małgosia rozważy każdy podzbiór monet jakie posiada, wyznaczy jego wartość v i liczność p i zapyta Jasia: „Jasiu, czy potrafisz wydać kwotę $k - v$?". Jeśli Jaś odpowie „Nie.", to Małgosia przechodzi dalej do następnego podzbioru. Jeśli Jaś odpowie „Tak, i to tylko q monetami: $a_7, a_{17}, \ldots$!", to razem będą umieli wydać kwotę k za pomocą $q + r$ monet i Małgosia może zapisać $q + r$ (i sumę odpowiednich podzbiorów monet) jako kandydata na odpowiedź.

Krótki przykład: niech $A = \{5, 3, 6, 2, 4, 8\}$ i $k = 16$. Jaś otrzymuje monety 5, 3, 6 i wylicza, że:

- kwotę 0 umie wydać za pomocą 0 monet;
- kwoty $3, 5, 6$ umie wydać za pomocą 1 monety;
- kwoty $8, 9, 11$ umie wydać za pomocą 2 monet;
- kwotę 14 umie wydać za pomocą 3 monet.

Małgosia otrzymuje monety 2, 4, 8 i rozpatruje kolejno podzbiory:

- $\emptyset$ – suma to 0, liczność to 0 i pytanie to: „Jasiu, czy potrafisz wydać kwotę 16?". Odpowiedź Jasia to „Nie.".
- $\{2\}$ – suma to 2, liczność to 1 i pytanie jest o kwotę 14. Odpowiedź Jasia to „Tak, i to tylko 3 monetami.", więc razem potrafią wydać docelową kwotę za pomocą 4 monet.
- $\{4\}$ – suma to 4, ale Jaś nie potrafi wydać kwoty 12.
- $\{2, 4\}$ – suma to 6, ale Jaś nie potrafi wydać kwoty 10.
- $\{8\}$ – suma to 8, a Jaś umie wydać kwotę 8 za pomocą 2 monet, zatem oboje potrafią wydać docelową kwotę za pomocą 3 monet.
- $\{2, 8\}$ – suma to 10, a Jaś umie wydać kwotę 6 jedną monetą, a więc w tym przypadku wynik to ponownie 3 monety.
- pozostałych podzbiorów Jaś nie jest w stanie uzupełnić do kwoty 16.

2.2 Analiza złożoności

Algorytm opisany wyżej ma dwie fazy – „fazę Jasia" i „fazę Małgosi".

Faza Jasia to wyznaczenie wszystkich kwot, jakie można wydać za pomocą pierwszej połowy zbioru monet i zapamiętanie tych informacji. Podzbiorów pierwszej połowy zbioru monet jest $2^{\lceil n/2 \rceil}$, a każdy podzbiór wymaga zsumowania $\mathcal{O}(n)$ nominałów, zatem faza Jasia ma złożoność czasową $\mathcal{O}(n \cdot 2^{\lceil n/2 \rceil})$

([1]), natomiast pamięciową $\mathcal{O}(2^{\lceil n/2 \rceil})$ – jedna komórka pamięci na każdą możliwą do wydania kwotę.

Faza Małgosi to rozważenie sumy nominałów każdego podzbioru posiadanych przez nią monet i zapytanie Jasia o dopełnienie tej kwoty do docelowej kwoty, tak więc Małgosia przejdzie przez $2^{\lfloor n/2 \rfloor}$ zbiorów i dla każdego wykona $\mathcal{O}(n)$ operacji (ponownie, przy odpowiedniej kolejności podzbiorów, Małgosia może wykonywać $\mathcal{O}(1)$ operacji na każdy podzbiór), zatem złożoność czasowa jej fazy wyniesie $\mathcal{O}(n \cdot 2^{\lfloor n/2 \rfloor})$.

3 Logartym dyskretny

3.1 Definicja problemu

Logarytm dyskretny z b przy podstawie a modulo p to taka liczba całkowita x, że zachodzi przystawanie

$$a^x \equiv b \pmod{p}.$$

Przykładowo, logarytmem dyskretnym z 13 przy podstawie 2 modulo 17 jest 6, ponieważ

$$2^6 = 64 = 51 + 13 = 3 \cdot 17 + 13 \equiv 13 \pmod{17}.$$

Nie zawsze jednak logarytm dyskretny istnieje – przykładowo logarytm dyskretny z 3 przy podstawie 4 modulo 5 nie istnieje, ponieważ 4 podniesiona do jakiejkolwiek potęgi nie da liczby o reszcie z dzielenia przez 5 równej 3.

Ogólnie uznaje się, że dla dużych p problem obliczenia logarytmu dyskretnego jest *trudny*.

3.2 Zastosowanie

Wspomniana wyżej domniemana *trudność* obliczenia logarytmu dyskretnego stanowi podstawę bezpieczeństwa algorytmu Diffiego–Hellmana, który służy do bezpiecznego przekazywania informacji przez publiczne łącza.

3.3 Baby-step giant-step

Jak wiemy z notatki o małym twierdzeniu Fermata, dla dowolnego p i a ciąg $a, a^2, a^3, \ldots$ jest okresowy z okresem co najwyżej $p - 1$.

W tym podrozdziale ponownie pomogą nam Jaś i Małgosia.

Tym razem Jaś policzy reszty z dzielenia przez p liczb $1 = a^0, a^1, a^2, \ldots, a^{\lfloor \sqrt{p} \rfloor - 1}$ i zapamięta zbiór par

$$\{(a^n \bmod p, n) : 0 \leqslant n < \lfloor \sqrt{p} \rfloor\}.$$

Małgosia natomiast rozważy reszty z dzielenia przez p liczb $1 = a^0, a^{\lfloor \sqrt{p} \rfloor}, a^{2\lfloor \sqrt{p} \rfloor}, \ldots, a^{\lceil \sqrt{p} \rceil \lfloor \sqrt{p} \rfloor}$ i dla każdej z tych liczb zapyta Jasia: „Jasiu, czy masz liczbę, która pomnożona przez moją (równą $a^{q\lfloor \sqrt{p} \rfloor}$) daje b modulo p?[2]”. Jeśli Jaś odpowie „Nie.”, to Małgosia przechodzi do kolejnej liczby. Jeśli Małgosia rozważała właśnie liczbę $a^{q \cdot \lfloor \sqrt{p} \rfloor}$ i Jaś odpowie „Tak, to a^r!”, to będzie wiadomo, że

$$a^{q \cdot \lfloor \sqrt{p} \rfloor + r} \equiv b \pmod{p},$$

a zatem $q \cdot \lfloor \sqrt{p} \rfloor + r$ jest szukanym logarytmem dyskretnym.

Ku pełności wywodu warto wspomnieć, że Jaś powinien użyć struktury `set`, `map` lub podobnej, by odpowiednio szybko zaspokajać ciekawość świata Małgosi.

3.4 Przykład

Zobaczmy jak Jaś i Małgosia obliczają logarytm dyskretny z 14 przy podstawie 3 modulo 17.

Najpierw zauważmy, że $\lfloor \sqrt{p} \rfloor = \lfloor \sqrt{17} \rfloor = 4$.

Jaś oblicza kolejne potęgi 3 modulo 17:

- $3^0 \equiv 1 \pmod{17}$;
- $3^1 \equiv 3 \pmod{17}$;
- $3^2 \equiv 9 \pmod{17}$;
- $3^3 \equiv 10 \pmod{17}$;

i zapamiętuje je wraz z odpowiednimi wykładnikami.

Małgosia rozważa kolejne potęgi skacząc wykładnikiem co 4:

[1] Okazuje się, że wszystkie podzbiory skończonego zbioru można ustawić w ciąg, którego sąsiednie elementy różnią się należeniem dokładnie jednego elementu oryginalnego zbioru, zatem tę złożoność można „zbić” o multiplikatywne n, wyznaczając sumy monet w podzbiorach w kolejności danej przez wspomniany ciąg podzbiorów.

[2] Jaś przeczytał notatkę o odwrotności modularnej i wie jak obliczyć szukaną przez Małgosię liczbę.

- $3^0 \equiv 1 \pmod{17}$, Małgosia szuka u Jasia liczby 14, bo $1 \cdot 14 \equiv 14 \pmod{17}$, ale Jaś odpowiada „Nie.”.
- $3^4 \equiv 13 \pmod{17}$, Małgosia szuka u Jasia liczby 5, bo $13 \cdot 5 \equiv 14 \pmod{17}$, ale Jaś odpowiada „Nie.”.
- $3^8 \equiv 16 \pmod{17}$, Małgosia szuka u Jasia liczby 3, bo $16 \cdot 3 \equiv 14 \pmod{17}$ i tym razem Jaś odpowiada „Tak, to 3^1!”, tak więc $3^{8+1} \equiv 14 \pmod{17}$ i odpowiedzią na całe zadanie jest 9.

3.5 Analiza złożoności

Faza Jasia ma złożoność czasową $\mathcal{O}(\sqrt{p} \log p)$, ponieważ rozważenie każdej z $\mathcal{O}(\sqrt{p})$ potęg wiąże się z dodaniem jednego elementu do struktury `set` lub `map`. Jaś musi jeszcze zapamiętać swój zbiór, na co poświęci $\mathcal{O}(\sqrt{p})$ pamięci.

Faza Małgosi ma taką samą złożoność czasową, ponieważ rozważa ona z grubsza tyle samo potęg a, co Jaś i każda jej potęga wyzwala pytanie, na które Jaś odpowiada w czasie $\mathcal{O}(\log p)$.

4 Zobacz też

- `https://en.wikipedia.org/wiki/Baby-step_giant-step`
- `https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange`