

Uniwersytet Warszawski
Wydział Matematyki, Informatyki i Mechaniki

Wiktor Zuba

Nr albumu: 320501

Efektywne algorytmy generacji obiektów kombinatorycznych???

Praca magisterska
na kierunku INFORMATYKA

Praca wykonana pod kierunkiem
prof. Wojciech Rytter
Instytut Informatyki

??? 2017

Oświadczenie kierującego pracą

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data

Podpis kierującego pracą

Oświadczenie autora (autorów) pracy

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data

Podpis autora (autorów) pracy

Streszczenie

???

Słowa kluczowe

???

Dziedzina pracy (kody wg programu Socrates-Erasmus)

???

Klasyfikacja tematyczna

???

Tytuł pracy w języku angielskim

Effective algorithms of combinatorial objects generation???

Spis treści

Wprowadzenie	5
1. Własności hiperkostki	7
1.1. Podstawowe definicje	7
1.1.1. Hiperkostka	7
1.1.2. Sąsiedztwo	8
1.1.3. Podgrafy	9
1.1.4. Graf z wadami	9
1.2. Podstawy kombinatoryczne	9
1.3. Własność ekspansji	10
2. Spójność wadliwej hiperkostki	13
2.1. Podejście ekspansywne	13
2.1.1. pseudokod i uwagi	14
2.2. Redukcja przy pomocy transformacji ścieżek	15
2.2.1. Transformacje ścieżek w hiperkostce	16
2.2.2. Dowód twierdzenia o lokalnej spójności	18
2.2.3. Algorytm	19
2.2.4. Analiza złożoności	21
2.3. Wnioski i zastosowania	21
2.3.1. Cykl Eulera	21
2.3.2. Długość ścieżki między dwoma punktami	21
2.3.3. Istnienie ścieżki między dwoma punktami	22
3. Długie ścieżki i cykle w grafie	23
3.1. Definicje	23
3.2. Długie ścieżki	23
3.3. Długie cykle	25
3.4. Długie pary ścieżek	25
4. Ścieżka i cykl Hamiltona	27
4.1. Kostka bez wadliwych wierzchołków	27
4.2. Ogólne podejście do problemu	27
4.3. Dowody dla szczególnych przypadków F	28
4.3.1. podkostka	28
4.3.2. Ścieżka izometryczna	29
4.3.3. Cykl izometryczny	30
4.3.4. Drzewo izometryczne	30

5. Największa odległość grafowa	33
5.1. Ścieżka Fibonacciego	33
5.2. Wyniki eksperymentalne	34
5.2.1. NMCS	35
5.2.2. Uwagi praktyczne	36
5.2.3. Uzyskane wartości	37
Bibliografia	39

Wprowadzenie

Rozdział 1

Własności hiperkostki

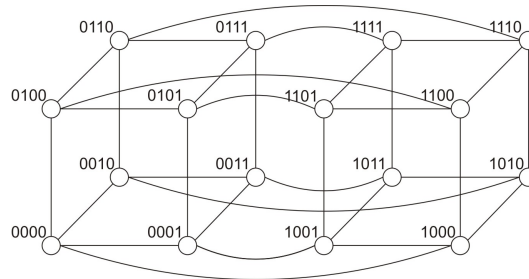
1.1. Podstawowe definicje

Definicja 1.1.1. Dla $n \in \mathbb{N}$ $[n] = \{0, \dots, n-1\}$ (zbiór pierwszych n liczb naturalnych).

1.1.1. Hiperkostka

Definicja 1.1.2. Hiperkostką wymiaru n (Q_n) nazwiemy graf, w którym każdy wierzchołek odpowiada ciągowi binarnemu długości n , zaś krawędzią połączone są te wierzchołki, których ciągi binarne różnią się na dokładnie jednej pozycji.

$$V(Q_n) = \{(v_0, \dots, v_{n-1}) : v_i \in \{0, 1\}\}, E(Q_n) = \{(u, v) : \sum_i |u_i - v_i| = 1\}$$



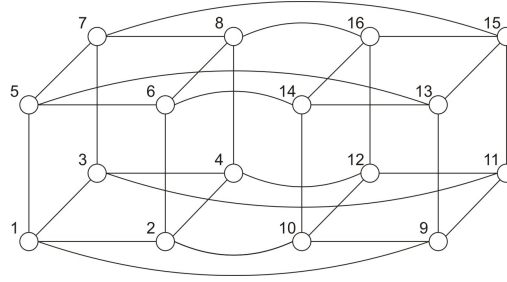
W przypadku pełnej hiperkostki bardzo łatwo jest określić długość najkrótszej ścieżki pomiędzy wierzchołkami – jest ona równa ilości pozycji na których różnią się ciągi tych wierzchołków.

Hiperkostka jest grafem dwudzielnym, w którym jedną składową jest zbiór wierzchołków o ciągach z parzystą liczbą jedynek, zaś drugą tych o ich nieparzystej liczbie.

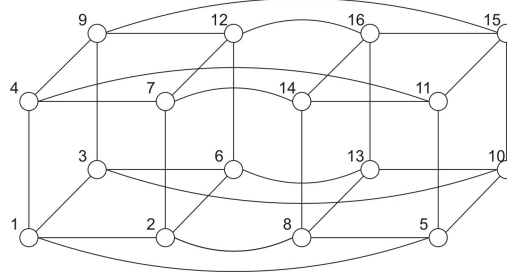
Co więcej przy badaniu hiperkostek często dzieli się je na $n+1$ warstw, gdzie dla $i \in [n+1]$ i -tą warstwę stanowią te wierzchołki, których ciągi binarne mają dokładnie i jedynek (warstwa zawiera zatem wierzchołki oddalone o i od wierzchołka zerowego ($\bar{0}$)).

Definicja 1.1.3. Dla dwóch wierzchołków hiperkostki definiujemy: $u\Delta v = \{i : u_i \neq v_i\}$, gdzie (u_0, \dots, u_{n-1}) i (v_0, \dots, v_{n-1}) to ciągi binarne wierzchołków u i v odpowiednio ($|u\Delta v|$ oznacza odległość wierzchołków w hiperkostce).

Definicja 1.1.4. Numerowaniem klasycznym (naturalnym) hiperkostki nazwiemy takie numerowanie $\varphi : V(Q_n) \rightarrow \{1, \dots, |V(Q_n)|\}$ jej wierzchołków, że $\varphi(v) = 1 + \sum_i v_i \cdot 2^i$



Definicja 1.1.5. Numerowaniem warstwowym hiperkostki nazwiemy jej numerowanie w kolejności przeszukiwania grafu wszerz zaczynając od wierzchołka $\bar{0}$ z wybieraniem sąsiadów w kolejności leksykograficznej.



Uwaga 1. Jest to takie numerowanie $\varphi : V(Q_n) \rightarrow \{1, \dots, |V(Q_n)|\}$ jej wierzchołków, że wierzchołki z i -tej warstwy otrzymują numery od $\sum_{j=0}^{i-1} \binom{n}{j} + 1$ do $\sum_{j=0}^i \binom{n}{j}$. W obrębie jednej warstwy numery przyznawane są przeciwnie do kolejności leksykograficznej na odwróconych słowach. $\varphi(v) > \varphi(u) \Leftrightarrow (\sum_{i=0}^n v_i > \sum_{i=0}^n u_i) \vee ((\sum_{i=0}^n v_i = \sum_{i=0}^n u_i) \wedge (\sum_{i=0}^n 2^{n-i} v_i < \sum_{i=0}^n 2^{n-i} u_i))$

Dowód. Indukcyjnie po warstwach.

Dla warstwy 0 oczywiste.

Zakładając, że i -ta warstwa jest ponumerowana w tym porządku weźmy dwa wierzchołki u, v z warstwy $i + 1$: $u = (\bar{y}_1, 1, \bar{x}), v = (\bar{y}_2, 0, \bar{x})$.

Jeśli \bar{y}_1 zawiera same 0, to \bar{y}_2 zawiera dokładnie jedną 1, sąsiedzi tych wierzchołków z poprzedniej warstwy o najmniejszych numerach to odpowiednio $(\bar{0}, 0, \bar{x}), (\bar{0}, 0, \bar{x})$, tak więc zostaną ponumerowane jako sąsiedzi tego samego wierzchołka, jednak u otrzyma mniejszy numer jako sąsiad mniejszy leksykograficznie.

Jeśli \bar{y}_1 zawiera 1, to \bar{y}_2 też, więc sąsiedzi tych wierzchołków z poprzedniej warstwy o najmniejszych numerach to odpowiednio $(\bar{y}'_1, 1, \bar{x}), (\bar{y}'_2, 0, \bar{x})$, gdzie \bar{y}'_1 i \bar{y}'_2 , to odpowiednio \bar{y}_1 i \bar{y}_2 z pierwszymi 1 zamienionymi na 0. Z założenia indukcyjnego sąsiad u ma mniejszy numer niż sąsiad v , więc u ma mniejszy numer niż v . \square

1.1.2. Sąsiedztwo

Definicja 1.1.6. Dla grafu G oraz wierzchołka $v \in V(G)$ definiujemy sąsiedztwo wierzchołka jako zbiór wierzchołków połączonych z nim krawędzią: $N(v) = \{u \in V(G) : (u, v) \in E(G)\}$.

Definicja 1.1.7. Dla grafu G oraz zbioru wierzchołków $S \subseteq V(G)$ definiujemy sąsiedztwo zbioru wierzchołków jako zbiór tych sąsiadów wierzchołków ze zbioru, które same do tego zbioru nie należą: $N(S) = (\bigcup_{v \in S} N(v)) \setminus S$

Definicja 1.1.8. Dla grafu G oraz zbioru wierzchołków $S \subseteq V(G)$ definiujemy wnętrze zbioru wierzchołków jako zbiór tych wierzchołków z S , których wszyscy sąsiedzi również należą do tego zbioru: $In(S) = \{v \in S : N(v) \subseteq S\}$

1.1.3. Podgrafy

Definicja 1.1.9. Dla danego $V \subseteq V(G)$ $G[V] = (V, \{uv \in E(G) : u, v \in V\})$ oznacza podgraf indukowany przez podzbiór wierzchołków V .

Definicja 1.1.10. Dla danego $V \subseteq V(G)$ $G - V = G[V(G) \setminus V]$ oznacza graf G z usuniętymi wierzchołkami V .

Definicja 1.1.11. Dla danego grafu G

$G^2 = (V(G), E(G) \cup \{uv : \exists w \in V(G) uw \in E(G) \cap vw \in E(G)\})$ oznacza kwadrat grafu, czyli graf z dodanymi krawędziami między wierzchołkami oddalonymi o co najwyżej 2.

1.1.4. Graf z wadami

Definicja 1.1.12. W grafie G możemy wyróżnić niektóre wierzchołki (czasem również krawędzie) i oznaczyć jako wadliwe. Graf z niepustym takim wyróżnionym zbiorem wierzchołków wadliwych $F \subseteq V(G)$ nazywamy grafem z wadami (lub grafem wadliwym)

Wadliwe wierzchołki (i/lub krawędzie) najczęściej traktowane są jako usunięte z grafu – mówimy w tym przypadku o grafie $G - F$. Wyróżnianie wadliwych wierzchołków w grafie zamiast definiowania nowego grafu jest umotywowane głównie w przypadkach, gdy pełny graf łatwo zdefiniować i zapisać w pamięci małej względem jego rozmiaru (np. klika, hiperkostka, graf de Bruijna), a zbiór wadliwych wierzchołków jest również mały.

1.2. Podstawy kombinatoryczne

$$\binom{2n}{n} = \frac{2^{2n} \Gamma(n + \frac{1}{2})}{\sqrt{\pi} \Gamma(n+1)}, \quad \binom{2n+1}{n} = \binom{2n+1}{n+1} = \frac{2^{2n+1} \Gamma(n + \frac{3}{2})}{\sqrt{\pi} \Gamma(n+2)}$$

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx \quad \text{dla } n \in \mathbb{N} \quad \Gamma(n) = (n-1)!,$$

$$\text{ogólniej dla } x \in \mathbb{R}, x > 1 \quad \frac{\Gamma(x+1)}{\Gamma(x)} = x, \quad \frac{\Gamma(x+\frac{1}{2})}{\Gamma(x)} < \frac{\Gamma(x+1)}{\Gamma(x+\frac{1}{2})} \Rightarrow \sqrt{x - \frac{1}{2}} < \frac{\Gamma(x+\frac{1}{2})}{\Gamma(x)} < \sqrt{x}$$

$$\text{Daje to ograniczenia: } \frac{2^{2n}}{\sqrt{\pi(n+\frac{1}{2})}} < \binom{2n}{n} < \frac{2^{2n}}{\sqrt{\pi n}}, \quad \frac{2^{2n+1}}{\sqrt{\pi(n+\frac{3}{2})}} < \binom{2n+1}{n} = \binom{2n+1}{n+1} < \frac{2^{2n+1}}{\sqrt{\pi(n+1)}}$$

$$\text{Lub równoważnie: } \frac{2^n}{\sqrt{\pi(\lfloor \frac{n}{2} \rfloor + \frac{1}{2})}} < \binom{n}{\lfloor \frac{n}{2} \rfloor} = \binom{n}{\lceil \frac{n}{2} \rceil} < \frac{2^n}{\sqrt{\pi \lceil \frac{n}{2} \rceil}}$$

Lemat 2. Dla $k \leq \lfloor \frac{n+1}{2} \rfloor$ zachodzi ograniczenie $\sum_{i=0}^{k-1} \binom{n}{i} \leq 2^{n-1} \binom{n}{\lfloor \frac{n}{2} \rfloor}$

Dowód. (Uogólnienie dowodu z podobnego lematu dla $n = 2m, k < m$ z [1])

Założmy najpierw, że $k < \lfloor \frac{n}{2} \rfloor$

$$\text{Zdefiniujmy } c = \frac{\binom{n}{k}}{\binom{n}{\lfloor \frac{n}{2} \rfloor}} < 1, t = \lfloor \frac{n}{2} \rfloor - k, A = \sum_{i=0}^{k-1} \binom{n}{i}, B = \sum_{i=k}^{\lfloor \frac{n}{2} \rfloor - 1} \binom{n}{i}$$

$$\forall 1 \leq i \leq k \quad \frac{\binom{n}{k-i}}{\binom{n}{\lfloor \frac{n}{2} \rfloor - i}} < \frac{\binom{n}{k-i+1}}{\binom{n}{\lfloor \frac{n}{2} \rfloor - i+1}} \Leftrightarrow \frac{k-i+1}{n-k+c} < \frac{\lfloor \frac{n}{2} \rfloor - c + 1}{\lfloor \frac{n}{2} \rfloor + c},$$

$$\text{co wynika z szeregu prostych nierówności } \frac{k-i+1}{n-k+c} \leq \frac{k-i+1}{k+c+1} < \frac{\lfloor \frac{n}{2} \rfloor - c + 1}{\lfloor \frac{n}{2} \rfloor + c+1} \leq \frac{\lfloor \frac{n}{2} \rfloor - c + 1}{\lfloor \frac{n}{2} \rfloor + c}$$

$$\text{Daje nam to ograniczenia } \forall 1 \leq i \leq k \quad \frac{\binom{n}{k-i}}{\binom{n}{\lfloor \frac{n}{2} \rfloor - i}} < c.$$

Suma ostatnich t wyrazów szeregu A jest majoryzowana przez $c \cdot B$, wcześniejszych t przez c razy suma ostatnich t (a więc przez $c^2 \cdot B$). Daje nam to oszacowanie $A < (c + c^2 + c^3 + \dots + c^{\lfloor \frac{k}{t} \rfloor}) \cdot B < (c + c^2 + c^3 + \dots) \cdot B = \frac{c}{1-c} \cdot B$. Jednocześnie $A + B = \sum_{i=0}^{\lfloor \frac{n}{2} \rfloor - 1} \binom{n}{i} < 2^{n-1}$.

$$A = c \cdot A + (1 - c) \cdot A = c(A + \frac{1-c}{c}A) < c \cdot (A + B) < c \cdot 2^{n-1}.$$

Pozostaje udowodnić przypadki większych k :

$$\text{Dla } n = 2m, k = m \sum_{i=0}^{m-1} \binom{2m}{i} = 2^{2m-1} - \frac{1}{2} \binom{2m}{m} < 2^{2m-1} = 2^{n-1} \cdot \frac{\binom{n}{k}}{\binom{\lfloor \frac{n}{2} \rfloor}{\lfloor \frac{n}{2} \rfloor}}$$

$$\text{Dla } n = 2m + 1, k = m \sum_{i=0}^{m-1} \binom{2m+1}{i} = 2^{2m} - \binom{2m+1}{m} < 2^{2m} = 2^{n-1} \cdot \frac{\binom{n}{k}}{\binom{\lfloor \frac{n}{2} \rfloor}{\lfloor \frac{n}{2} \rfloor}}$$

$$\text{Dla } n = 2m + 1, k = m + 1 \sum_{i=0}^m \binom{2m+1}{i} = 2^{2m} = 2^{n-1} \cdot \frac{\binom{n}{k}}{\binom{\lfloor \frac{n}{2} \rfloor}{\lfloor \frac{n}{2} \rfloor}} \text{ (jedyna nieostra nierówność)}$$

□

1.3. Własność ekspansji

Definicja 1.3.1. Graf G posiada własność ε -ekspansji wierzchołkowej, jeżeli dla każdego zbioru wierzchołków $S \subseteq V(G)$ takiego, że $|S| \leq \frac{|V(G)|}{2}$ zachodzi $|N(S)| \geq \varepsilon \cdot |S|$

Lemat 3. Zbiór pierwszych l wierzchołków hiperkostki według numerowania warstwowego posiada maksymalne wnętrze wśród zbiorów wielkości l .

Jest to jeden z lematów dowodzonych w pracy [2].

Lemat 4. Dla hiperkostki do udowodnienia własności ε_n -ekspansji wierzchołkowej wystarczy rozważyć zbiory S postaci $S_k, k \leq 2^{n-1}$.

Dowód. Weźmy dowolne $S \subseteq V(G), l = |S| + |N(S)|$ z Lematu 1.3 wynika, że $\frac{|N(S)|}{|S|} = \frac{|N(S)| + |S|}{|S|} - 1 \geq \frac{|S_l|}{|In(S_l)|} - 1 = \frac{|S_l \setminus In(S_l)|}{|In(S_l)|} \geq \frac{|N(In(S_l))|}{|In(S_l)|}$. Z definicji S_l wynika, że $In(S_l) = S_k$ dla $k = In(S_l)$.

Pozostaje udowodnić, że wystarczy rozważyć te S_k , że $k \leq 2^{n-1}$

Dla $l = |N(S)| + |S| \geq (\varepsilon_n + 1) \cdot 2^{n-1}$ mamy $|S| > 2^{n-1}$ lub $|N(S)| \geq \varepsilon_n |S|$, wystarczy więc rozważyć przypadek $l < (\varepsilon_n + 1) \cdot 2^{n-1}$.

Dla $n = 2m + 1$ weźmy $k = 2^{n-1} = \sum_{i=0}^m \binom{2m+1}{i}$, wtedy S_k = pełne $m + 1$ pierwszych warstw i $N(S_k)$ = warstwa $m + 1$. Przykład ten pokazuje, że $\varepsilon_n \leq \frac{\binom{2m+1}{m+1}}{2^{2m}}$, więc $l < 2^{2m} + \binom{2m+1}{m+1} \Rightarrow S_l$ mieści się w pierwszych $m + 2$ warstwach $\Rightarrow S_k = In(S_l)$ mieści się w pierwszych $m + 1$ warstwach $\Rightarrow k \leq 2^{2m} = 2^{n-1}$.

Dla $n = 2m$ weźmy $k = 2^{n-1} = \sum_{i=0}^{m-1} \binom{2m}{i} + \frac{1}{2} \binom{2m}{m}$, wtedy S_k = pełne m pierwszych warstw + połowa środkowej. W środkowej warstwie pierwsze $\binom{2m-1}{m-1} = \frac{1}{2} \binom{2m}{m}$ wierzchołków to dokładnie te, których ciągi binarne kończą się na 1. Wtedy też $S_k \cup N(S_k)$ to dokładnie pełne $m + 1$ pierwszych warstw plus te wierzchołki z warstwy $m + 2$, które kończą się na 1 $\Rightarrow |N(S_k)| = \binom{2m-1}{m-1} + \binom{2m-1}{m} = \binom{2m-1}{m-1} + \binom{2m-1}{m} = \binom{2m}{m}$. Przykład ten pokazuje, że $\varepsilon_n \leq \frac{\binom{2m}{m}}{2^{2m-1}}$, więc $l < 2^{2m-1} + \binom{2m}{m} \Rightarrow S_l$ mieści się w pierwszych $m + 1$ warstwach plus tych wierzchołkach z warstwy $m + 2$, które kończą się na 1 $\Rightarrow k \leq 2^{n-1}$. □

Wniosek 5. Hiperkostka wymiaru n nie posiada własności $\frac{2\sqrt{2}}{\sqrt{\pi n}}$ -ekspansji wierzchołkowej.

Dowód. Dla $n = 2m + 1$

$$\frac{|N(S_{2^{2m}})|}{|S_{2^{2m}}|} = \frac{\binom{2m+1}{m+1}}{2^{2m}} = \frac{2^{2m+1}}{2^{2m} \sqrt{\pi(m+1)}} = \frac{2}{\sqrt{\pi(m+1)}} = \frac{2}{\sqrt{\pi(\frac{n}{2} + \frac{1}{2})}} = \frac{2\sqrt{2}}{\sqrt{\pi(n+1)}} < \frac{2\sqrt{2}}{\sqrt{\pi n}}.$$

$$\text{Dla } n = 2m \frac{|N(S_{2^{2m-1}})|}{|S_{2^{2m-1}}|} = \frac{\binom{2m}{m}}{2^{2m-1}} < \frac{2^{2m}}{2^{2m-1} \sqrt{\pi(m+1)}} = \frac{2}{\sqrt{\pi m}} = \frac{2}{\sqrt{\pi \cdot \frac{n}{2}}} = \frac{2\sqrt{2}}{\sqrt{\pi n}}. \quad \square$$

Twierdzenie 6. Hiperkostka Q_n posiada własność $\frac{1}{\sqrt{\pi n}}$ -ekspansji wierzchołkowej.

Dowód. Jeśli $k = \sum_{i=0}^r \binom{n}{i}$ (pełne $r+1 \leq \lfloor \frac{n}{2} \rfloor + 1$ warstw), to $\frac{|N(S_k)|}{|S_k|} = \frac{\binom{n}{r+1}}{\sum_{i=0}^r \binom{n}{i}} > \frac{\binom{n}{r+1} \binom{n}{\lfloor \frac{n}{2} \rfloor}}{2^{n-1} \binom{n}{r+1}} = \frac{\binom{n}{\lfloor \frac{n}{2} \rfloor}}{2^{n-1}} > \frac{2^n}{2^{n-1} \sqrt{\pi(\lfloor \frac{n}{2} \rfloor + \frac{1}{2})}} = \frac{2}{\sqrt{\pi(\lfloor \frac{n}{2} \rfloor + \frac{1}{2})}} \geq \frac{2\sqrt{2}}{\sqrt{\pi(n + \frac{3}{2})}} \geq \frac{2}{\sqrt{\pi n}}$ (dla $n \geq 2$).

($n = 2m + 1, r = m$ rozważone w 4)

Jeśli $k = \sum_{i=0}^r \binom{n}{i} + \binom{n-1}{r}$ (pełne $r+1 \leq \lfloor \frac{n}{2} \rfloor + 1$ warstw plus te wierzchołki z warstwy $r+2$, których ciągi binarne kończą się na 1).

$$|N(S_k) \cup S_k| = \sum_{i=0}^{r+1} \binom{n}{i} + \binom{n-1}{r+1} \Rightarrow |N(S_k)| = 2 \cdot \binom{n-1}{r+1}$$

$$\begin{aligned} \frac{|N(S_k)|}{|S_k|} &= \frac{2 \cdot \binom{n-1}{r+1}}{\sum_{i=0}^r \binom{n}{i} + \binom{n-1}{r}} > \frac{2 \cdot \binom{n-1}{r+1} \binom{n}{\lfloor \frac{n}{2} \rfloor}}{2^{n-1} \cdot (\binom{n}{r+1} + \binom{n-1}{r}) \cdot \binom{n}{\lfloor \frac{n}{2} \rfloor}} = \frac{2 \cdot \binom{n-1}{r+1} \binom{n}{\lfloor \frac{n}{2} \rfloor}}{2^{n-1} \cdot (\binom{n-1}{r} + \binom{n-1}{r+1}) + \binom{n-1}{r} \cdot \binom{n}{\lfloor \frac{n}{2} \rfloor}} > \\ \frac{2 \cdot \binom{n-1}{r+1} \binom{n}{\lfloor \frac{n}{2} \rfloor}}{2^{n-1} \cdot (\binom{n-1}{r+1} + \binom{n-1}{r}) \cdot \binom{n}{\lfloor \frac{n}{2} \rfloor}} &= \left(\frac{2^{n-1}}{\binom{n}{\lfloor \frac{n}{2} \rfloor}} + \frac{\binom{n-1}{r}}{2 \cdot \binom{n-1}{r+1}} \right)^{-1} > \left(\frac{\sqrt{\pi(n + \frac{3}{2})}}{2\sqrt{2}} + \frac{1}{2} \right)^{-1} = \frac{2\sqrt{2}}{\sqrt{\pi(n + \frac{3}{2})} + \sqrt{2}} > \frac{2}{\sqrt{\pi n}} \text{ (dla } \\ n \geq 7). \end{aligned}$$

W pozostałych przypadkach można otrzymać ograniczenie choć dużo gorsze wiedząc, że dodanie wierzchołka do S zmniejszy $N(S)$ o co najwyżej 1.

Weźmy teraz $\sum_{i=0}^r \binom{n}{i} < k < \sum_{i=0}^r \binom{n}{i} + \binom{n-1}{r}$

$$\frac{|N(S_k)|}{|S_k|} > \frac{\binom{n}{r+1} - \binom{n-1}{r}}{\sum_{i=0}^r \binom{n}{i} + \binom{n-1}{r}} = \frac{\binom{n-1}{r+1}}{\sum_{i=0}^r \binom{n}{i} + \binom{n-1}{r}} \geq \frac{\frac{1}{2} \binom{n}{r+1}}{\sum_{i=0}^r \binom{n}{i} + \frac{1}{2} \binom{n}{r+1}} \left(\frac{\sqrt{\pi(n + \frac{3}{2})}}{\sqrt{2}} + 1 \right)^{-1} > \frac{1}{\sqrt{\pi n}} \text{ (dla } n \geq 7).$$

Analogicznie dla $\sum_{i=0}^r \binom{n}{i} + \binom{n-1}{r} < k < \sum_{i=0}^{r+1} \binom{n}{i}$

$$\begin{aligned} \frac{|N(S_k)|}{|S_k|} &> \frac{2 \binom{n-1}{r+1} - \binom{n-1}{r}}{\sum_{i=0}^{r+1} \binom{n}{i}} = \frac{\binom{n-1}{r+1}}{\sum_{i=0}^{r+1} \binom{n}{i}} > \frac{\binom{n-1}{r+1}}{2^{n-1} \left(\frac{\binom{n}{r+1}}{\binom{n}{\lfloor \frac{n}{2} \rfloor}} + \binom{n}{r+1} \right)} \geq \frac{\binom{n}{r+1} \binom{n}{\lfloor \frac{n}{2} \rfloor}}{(2^{n-1} + \binom{n}{\lfloor \frac{n}{2} \rfloor}) \binom{n}{r+1}} = \frac{\binom{n}{\lfloor \frac{n}{2} \rfloor}}{2^{n-1} + \binom{n}{\lfloor \frac{n}{2} \rfloor}} > \\ \left(\frac{\sqrt{\pi(n + \frac{3}{2})}}{\sqrt{2}} + 1 \right)^{-1} &> \frac{1}{\sqrt{\pi n}} \text{ (dla } n \geq 7). \end{aligned}$$

Dla przypadków $n \leq 6$ można ręcznie sprawdzić wszystkie 2^{n-1} przypadków, aby również otrzymać oszacowanie $\frac{1}{\sqrt{\pi n}}$. \square

Rozdział 2

Spójność wadliwej hiperkostki

Ten rozdział jest napisany w większości na podstawie [3].

Uwaga 7. Aby zbadać spójność grafu $G - F$ dla spójnego grafu G wystarczy sprawdzić czy wciąż istnieje ścieżka pomiędzy dowolnymi dwoma wierzchołkami, które oryginalnym grafie sąsiadowały z jakimś spośród usuniętych wierzchołków (wszystkie takie wierzchołki należą do jednej spójnej składowej).

Dowód. Aby udowodnić spójność trzeba pokazać, że istnieje ścieżka pomiędzy dowolnymi dwoma wierzchołkami, jednak skoro w oryginalnym grafie taka ścieżka istniała, to w nowym grafie jedyną przeszkodą jest to, że na tej ścieżce mogły występować wierzchołki, które zostały usunięte. Taką ścieżkę można naprawić wstawiając w miejsca od pierwszego do ostatniego wystąpienia wierzchołka usuniętego ścieżkę pomiędzy odpowiednimi ich sąsiadami istniejącą w pomniejszonym grafie. \square

2.1. Podejście ekspansywne

Twierdzenie 8. Niech graf G posiada własność ε -ekspansji wierzchołkowej z $\varepsilon > 0$ i maksymalny stopień wierzchołka Δ , oraz dana jest wyrocznia zwracająca dla danego wierzchołka listę jego sąsiadów. Wtedy istnieje algorytm, który otrzymuje na wejściu zbiór $F \subseteq V(G)$ oraz ε i testuje spójność $G - F$ w czasie $O\left(\frac{|F|^2 \cdot \Delta^2 \cdot \log(|V(G)|)}{\varepsilon}\right)$

Lemat 9. Spójna składowa $S \subseteq V(G) \setminus F$ grafu $G - F$ jest jednego z dwóch typów:

- główna – $|S| > \frac{|V(G)|}{2}$
- mała – $|S| \leq \frac{|F|}{\varepsilon}$

Uwaga 10. Co prawda dla dużego $|F|$ i małego ε może być tak, że składowa jest jednocześnie główna i mała, jednak po pierwsze jest to przypadek mało interesujący, gdyż wtedy zwykle przeszukiwanie grafu spełnia tezę twierdzenia, a po drugie przypadek ten nie psuje w żaden sposób otrzymywanego algorytmu. W lemacie istotne jest to, że w grafie nie ma składowych średnich wielkości.

Fakt 11. Może być tylko jedna składowa główna.

Dowód. (Lematu)

Weźmy spójną składową S grafu $G - F$ ($N_{G-F}(S) = 0$), jeżeli $|S| \leq \frac{|V(G)|}{2}$, to z własności ε -ekspansji wierzchołkowej grafu G $|N_G(S)| \geq \varepsilon \cdot |S|$ (gdzie S jest teraz traktowane jako podzbiór

wierzchołków grafu G). Gdyby zachodziło $|S| > \frac{|F|}{\varepsilon}$, to mielibyśmy $|N_G(S)| > \frac{\varepsilon \cdot |F|}{\varepsilon} = |F|$, co daje sprzeczność ponieważ aby w grafie $G - F$ to sąsiedztwo było puste z grafu G trzeba usunąć co najmniej $N_G(S)$ wierzchołków. \square

Dowód. (Twierdzenia)

Chcemy sprawdzić, czy wszyscy sąsiedzi wierzchołków usuniętych należą do tej samej spójnej składowej. Na podstawie lematu 9, jeśli składowa zawierająca taki wierzchołek jest większa niż $\frac{|F|}{\varepsilon}$, to jest to składowa główna. Jeżeli wszystkie takie wierzchołki spełniają ten warunek, to $G - F$ jest spójny. Jeżeli natomiast, któraś z tych składowych okaże się mała, to $G - F$ nie jest spójny.

Wystarczy więc uruchomić liniowe przeszukiwanie grafowe w każdym wierzchołku sąsiadującym z wierzchołkiem wadliwym i przerywać po przejrzaniu $\frac{|F|}{\varepsilon}$ wierzchołków.

Algorytm liniowego przeszukiwania grafowego uruchamiany jest co najwyżej $|F| \cdot \Delta$ razy. Za każdym razem przeglądamy co najwyżej $\frac{|F|}{\varepsilon}$ wierzchołków. Dla każdego przeglądanej wierzchołka sprawdzamy co najwyżej Δ sąsiadów, a odpowiedź wyroczni zajmuje $O(\log(|V(G)|))$ czasu. Daje to złożoność z tezy twierdzenia. \square

Wniosek 12. *Ponieważ zgodnie z twierdzeniem 6 hiperkostka Q_n posiada własność $\frac{1}{\sqrt{\pi n}}$ -ekspansji wierzchołkowej, oraz można znaleźć wszystkich sąsiadów wierzchołka w czasie liniowym od ich ilości powyższy algorytm testuje spójność wadliwej hiperkostki w czasie $O(|F|^2 \cdot n^{3.5})$ (wyrażonego w ilości operacji arytmetycznych).*

Uwaga 13. *Ze względu na długość zapisu identyfikatora wierzchołka liniową od wymiaru hiperkostki nie da się przeprowadzać operacji na wierzchołkach w czasie szybszym niż n . To dolne ograniczenie jest osiągalne przy przechowywaniu przejrzanych wierzchołków w hashmapie (czas oczekiwany operacji $O(n)$, złożoność pamięciowa całej struktury $O(n^{0.5}|F|)$), lub w drzewie prefiksowym (czas pesymistyczny operacji $O(n)$, złożoność pamięciowa całej struktury $O(n^{1.5}|F|)$). Pozwala to w łatwy sposób uzyskać efektywną wyrocznię, a więc i algorytm o złożoności z wniosku.*

2.1.1. pseudokod i uwagi

W algorytmie wykorzystywana jest struktura T z operacjami

- $Insert(v, T)$ wstawiającą wierzchołek v do struktury T
- $Retrieve(v, T)$ zwracającą binarną informację o obecności wierzchołka v w strukturze T

które wymagają $O(n)$ czasu na wykonanie (jak w uwadze 13).

Przeszukiwanie grafowe odbywa się przy pomocy funkcji o pseudokodzie:

```
DFS(v){
    counter ++;
    Insert(v, T);
    if(counter ≥ size) return(TRUE);
    foreach(u ∈ N(v)){
        if(Retrieve(u, T) == FALSE){
            if(DFS(u)) return(TRUE);
        }
    }
}
```



```

    return(FALSE);
}

```

Spójność sprawdzana jest przy pomocy funkcji głównej o pseudokodzie:

```

Conectivity(n, F){
    T2 = empty_structureT();
    counter = 0;
    foreach(f ∈ F){
        Insert(f, T2);
        counter ++;
    }
    size = sqrt( $\pi * n$ ) * counter;
    foreach(f ∈ F){
        foreach(v ∈ N(f)){
            if(Retrieve(v, T2) == FALSE){
                counter = 0;
                T = T2;
                if(DFS(v) == FALSE) return(FALSE);
            }
        }
    }
    return(TRUE);
}

```

Uwaga 14. Aby przeiterować po $N(v)$ wystarczy przeiterować się po współrzędnych uzyskując sąsiada poprzez zanegowanie tej współrzędnej w zapisie binarnym v .

Uwaga 15. Można użyć dodatkowej struktury T w której przechowywane są wszystkie wierzchołki z poprzednich wywołań $DFS(v)$ z funkcji głównej. Wtedy przy kolejnych użyciach $DFS(v)$ można sprawdzać, czy wierzchołek nie był już wcześniej w jakiejś składowej (można wtedy od razu zwrócić $TRUE$). Teoretycznie może to zwiększyć złożoność dwukrotnie, jednak w praktyce będzie to dużo szybsze (już nawet z tego względu, że albo inni sąsiedzi tego samego f są oddaleni o 2, albo na drodze staje inny wierzchołek z F), w szczególności przy użyciu bardziej wyszukanych kolejności przeszukiwania (np. próba dojścia do wierzchołka $\bar{0}$).

2.2. Redukcja przy pomocy transformacji ścieżek

Algorytm przedstawiony w poprzednim podrozdziale jest dowodem na to, że testowanie spójności wadliwej hiperkostki może być zrobione wielomianowo ze względu na ilość wad i wymiar hiperkostki. Algorytm ten wykorzystuje jednak bardzo płytko potencjał tak regularnego grafu. W tym podrozdziale przedstawię algorytm, który dzięki głębszemu wykorzystaniu własności hiperkostki otrzymuje lepsze rezultaty złożonościowe.

Definicja 2.2.1. Na potrzeby tego podrozdziału definiuję ze pracą [3] dla $F \subseteq V(Q_n)$ podgraf $G(F) = (A \cup B \cup F, E)$ grafu Q_n , gdzie $A = N(F)$, $B = N(A) \setminus F$, $E = \{uv \in E(Q_n) : u \in A \cup F\}$ (podgraf zawierający wierzchołki w odległości ≤ 2 od wierzchołków wadliwych, plus krawędzie w których jeden z końców jest wadliwy lub z takim sąsiaduje).

Twierdzenie 16. Dla $F \subseteq V(Q_n)$ graf $Q_n - F$ jest spójny wtedy i tylko wtedy gdy dla każdej C – spójnej składowej $Q_n^2[F]$ spójny jest graf $G(C) - C$.

2.2.1. Transformacje ścieżek w hiperkostce

Definicja 2.2.2. Dla ścieżki $W = (v_0, v_1, \dots, v_n)$ (z możliwymi powtórzeniami) w hiperkostce sekwencją tranzycji nazywamy ciąg $\tau = (d_1, d_2, \dots, d_n)$, gdzie d_i jest współrzędną na której różni się ciągi binarne wierzchołków v_{i-1} i v_i .

Fakt 17. τ jest sekwencją tranzycji pewnej uv -ścieżki w Q_n wtedy i tylko gdy $u\Delta v = \{i \in [n] : \#(\tau, i) \text{ nieparzyste}\}$, gdzie $\#(\tau, i)$ to ilość wystąpień i w sekwencji τ .

Dla τ – sekwencji tranzycji uv -ścieżki W definiujemy trzy operacje:

- $\text{swap}(\tau_1, i, j, \tau_2) = (\tau_1, j, i, \tau_2)$ dla $\tau = (\tau_1, i, j, \tau_2)$
- $\text{insert}_i(\tau_1, \tau_2) = (\tau_1, i, i, \tau_2)$ dla $\tau = (\tau_1, \tau_2), i \in [n]$
- $\text{delete}(\tau_1, i, i, \tau_2) = (\tau_1, \tau_2)$ dla $\tau = (\tau_1, i, i, \tau_2)$

Wszystkie te operacje nie zmieniają parzystości wystąpień współrzędnych, dlatego też dowolnie w ten sposób zmodyfikowana sekwencja wciąż jest sekwencją tranzycji pewnej uv -ścieżki w Q_n .

Definicja 2.2.3. Dwie ścieżki, których sekwencje tranzycji τ, ρ spełniają $\forall_{i \in [n]} \#(\tau, i) = \#(\rho, i)$ nazywamy równoważnymi.

Uwaga 18. Dla dowolnych dwóch uv -ścieżek w Q_n istnieje sekwencja operacji $\text{swap}, \text{insert}, \text{delete}$ (w tej kolejności bez przeplotów), która przemienia sekwencję tranzycji pierwszej w sekwencję tranzycji drugiej.

Dowód. Jeśli dwie ścieżki są równoważne, to można jedną przekształcić w drugą przy pomocy samych operacji swap .

W przypadku gdy sekwencje mają różne liczności wystąpień współrzędnych, to można je doprowadzić do takich τ', ρ' , że $\forall_{i \in [n]} \#(\tau', i) = \#(\rho', i)$ przy pomocy samych operacji insert (używanych w dowolnie wybranych wierzchołkach). \square

Definicja 2.2.4. Na potrzeby dowodu twierdzenia 16 dla uv -ścieżki $W = (w_0, w_1, \dots, w_k)$ (gdzie $w_0 = u, w_k = v$) wierzchołek w_i nazywamy portem, jeśli nie jest wierzchołkiem wadliwym, ale dokładnie jeden z jego sąsiadów w ścieżce należy do F (port musi więc należeć do A).

W przypadku tej definicji należy rozróżnić przeplatające się pojęcia wierzchołka grafu i jego wystąpienia na ścieżce – portem nazywane jest konkretne wystąpienie na ścieżce, inne jego wystąpienia nie muszą być portami.

Dla C spójnej składowej $G(F) - F$ przez $p(C, W)$ oznaczamy ilość portów w części W należącej do C .

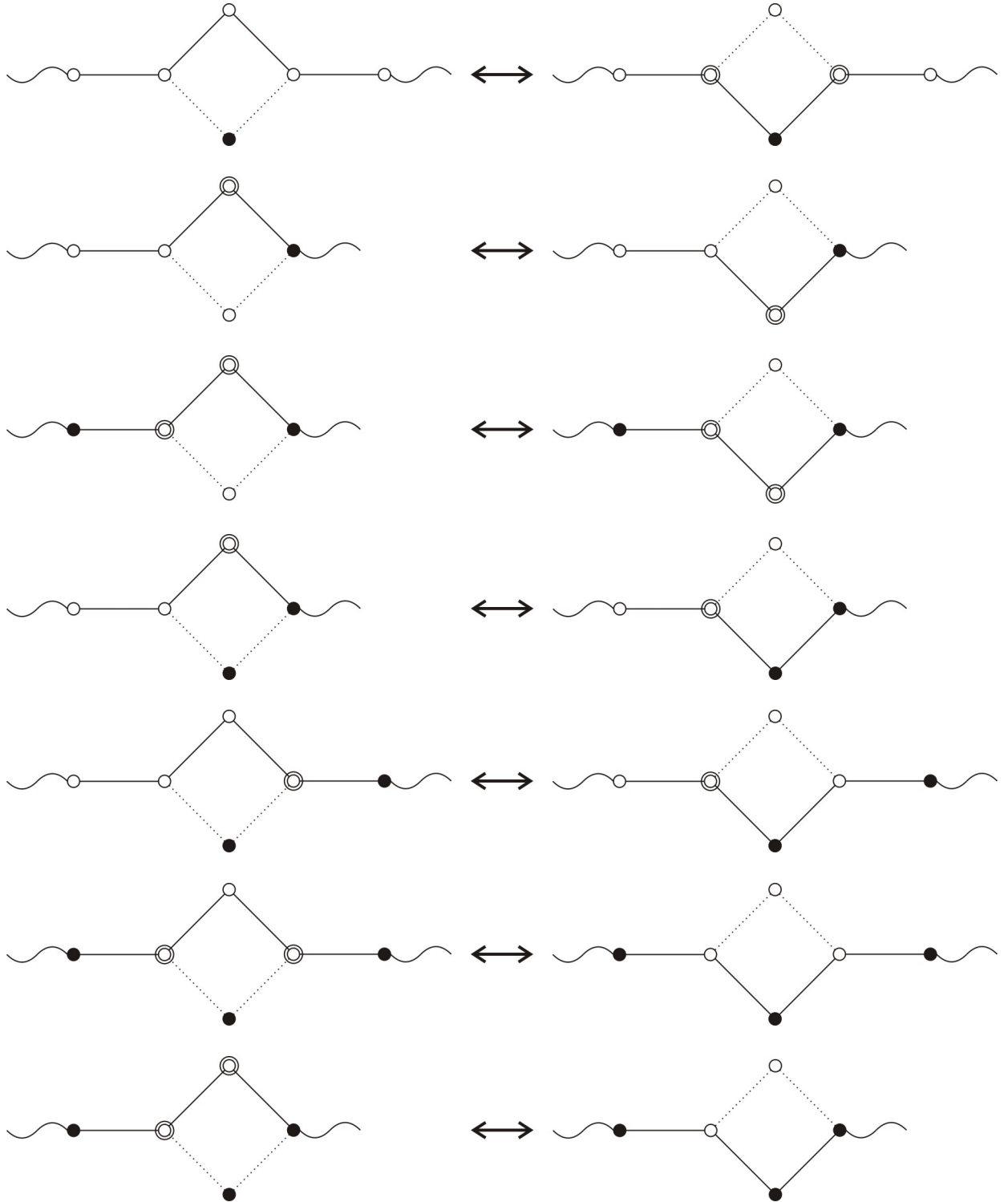
Lemat 19. Operacja swap zachowuje parzystość $p(C, W)$.

Dowód. Dowód stanowi rysunkowe rozpatrzenie wszystkich możliwych przypadków w których w wyniku operacji swap powstaje i/lub znika pewien port (przypadki przy końcach ścieżki można "dopełnić" zwykłymi wierzchołkami do przypadków ze środka ponieważ wierzchołki końcowe nie są wadliwe).

○ – wierzchołek zwykły
 ——— – krawędź ścieżki

● – wierzchołek wadliwy
 – krawędź spoza ścieżki

⊙ – port
 ~~~~~ – nieistotna reszta ścieżki



Wszystkie rozrysowane tu wierzchołki należą do  $G(F)$ , a rozrysowane części niewadliwe two-

rzą podgraf spójny, dlatego też wszystkie te zmiany odbywają się w jednej spójnej składowej  $G(F) - F$ . Za każdym razem ilość portów zmienia się o 0 lub 2, a więc parzystość pozostaje bez zmian.  $\square$

### 2.2.2. Dowód twierdzenia o lokalnej spójności

**Lemat 20.** *Niech  $F \subseteq V(Q_n)$  takie, że  $G(F)$  jest spójne. Jeśli  $Q_n - F$  jest spójne, to  $G(F) - F$  również.*

*Dowód.* Załóżmy przeciwnie – istnieją wierzchołki  $u, v \in A \cup B = V(G(F) - F)$  dla których istnieje ścieżka  $P$  w  $Q_n - F$ , ale nie ma takiej w  $G(F) - F$ . Skoro w  $G(F) - F$  nie ma takiej ścieżki, to w  $P$  musi występować wierzchołek  $x$  spoza  $A \cup B$ .

$G(F)$  jest spójne, więc musi istnieć też druga ścieżka  $R$  łącząca  $u$  z  $v$  w tym właśnie grafie, na której występuje wierzchołek  $y \in F$ .

Podobnie jak w dowodzie lematu 18 ścieżki te mogą być napompowane w wierzchołkach  $x$  i  $y$  odpowiednio sekwencjami operacji *insert*.

Ponieważ  $x$  jest oddalone od  $F$  dodane do ścieżki  $P$  wierzchołki nie uczynią z żadnego wystąpienia  $x$  portu i same również nie staną się portami. Ponieważ  $y$  należy do  $F$  dodane do ścieżki  $R$  wierzchołki będą miały dokładnie dwóch sąsiadów z  $F$  (tego samego dwukrotnie), a więc nie będą portami.

Oznacza to, że uzyskane ścieżki równoważne mają tyle samo portów we wszystkich spójnych składowych  $G(F) - F$  co odpowiadające im nieprzekształcone a na podstawie lematów 18 i 19 ich parzystości między sobą się zgadzają (czyli zgadzają się dla  $P$  i  $R$ ). Na ścieżce  $P$  nie ma żadnych portów ponieważ nie występuje w niej żaden wierzchołek wadliwy, daje to sprzeczność ponieważ dla spójnych składowych  $G(F) - F$  w której występują  $u$  i  $v$  ścieżka  $R$  ma nieparzyste ilości portów (np. można dobrać taką ścieżkę, która wchodzi do/opuszcza składowe co najwyżej raz).  $\square$

**Wniosek 21.** *Dla  $F \subseteq V(Q_n)$  takiego, że  $Q_n^2[F]$  jest spójne ze spójności  $Q_n - F$  wynika spójność  $G(F) - F$ .*

*Dowód.* Jeśli dwa wierzchołki  $Q_n^2[F]$  są połączone, to w oryginalnym grafie musiały być w odległości  $\leq 2$ , a więc w  $G(F)$  muszą być połączone albo bezpośrednio albo poprzez pojedynczy wierzchołek z  $A$ , a więc graf  $G(F)$  jest spójny co sprawia, że spełnione są założenia lematu.  $\square$

**Lemat 22.** *Niech  $F \subseteq V(Q_n)$  taki, że  $G(C) - C$  jest spójne dla każdej  $C$  spójnej składowej  $Q_n^2[F]$ . Wtedy  $Q_n - F$  również jest spójny.*

*Dowód.* Dla dowolnie wybranych dwóch wierzchołków  $u, v \in V(Q_n - F)$  weźmy  $W$  – ścieżkę między nimi w pełnym  $Q_n$ . Jeśli  $W$  nie zawiera wadliwego wierzchołka, to jest poprawną ścieżką w  $Q_n - F$ . W przeciwnym przypadku znajdujemy na tej ścieżce pierwsze wystąpienie wierzchołka wadliwego. Poprzedni wierzchołek na ścieżce oraz pierwszy kolejny z poza zbioru  $F$  są dwoma niewadliwymi wierzchołkami należącymi do  $G(C)$ , gdzie  $C$  jest spójną składową  $Q_n^2[F]$  (oddalone o 1 od wadliwych wierzchołków, które są połączone ścieżką samych wadliwych wierzchołków). W  $G(C)$  nie ma wadliwych wierzchołków spoza  $C$ , ponieważ oznaczałoby to, że taki wierzchołek jest oddalony o  $\leq 2$  od pewnego wierzchołka z  $C$ , a więc byłby z nim połączony w  $Q_n^2$ , dlatego też ścieżka ze spójnego założenia  $G(C) - C$  nie zawiera wadliwego wierzchołka. Wystarczy więc wadliwą część ścieżki  $W$  zastąpić odpowiednią ścieżką z  $G(C) - C$  aby otrzymać poprawną ścieżkę w  $Q_n - F$ .  $\square$

*Dowód.* (Twierdzenia 16)

Lemat 22 jest implikacją w jedną stronę. W drugą stronę dla spójnego  $Q_n^2[F]$  jest dana wnioskiem 21. Wystarczy udowodnić, że nic nie psuje się w przypadku gdy  $Q_n^2[F]$  ma więcej niż jedną spójną składową. Dla  $C$  – spójnej składowej  $Q_n^2[F]$  jeśli  $Q_n - F$  jest spójne, to jest takie również  $Q_n - C$  (dla wierzchołków spoza  $Q_n - F$  te same ścieżki są dobre, dla tych z  $F \setminus C$  dowolny sąsiad należy do  $Q_n - F$ , więc również łatwo zbudować ścieżkę), a więc spójne jest również  $G(C) - C$  co kończy dowód.  $\square$

### 2.2.3. Algorytm

Stosując powyższe twierdzenie można uzyskać wielomianowy algorytm używając jedynie przeszukiwania grafowego podobnie jak w podrozdziale 2.1. Można jednak uzyskać lepsze rezultaty używając dodatkowo struktury *Find-Union* i sprawdzając spójności już w trakcie budowania podgrafów  $G(C) - C$ .

W algorytmie używane są:

- struktura *Find-Union*  $D$  z operacjami:
  - $Make(v, D)$  tworzącą singleton  $\{v\}$
  - $Find(v, D)$  zwracającą wskaźnik na zbiór zawierający  $v$
  - $Union(u, v, D)$  łączącą zbiór zawierający  $u$  ze zbiorem zawierającym  $v$

których zamortyzowany czas można ograniczyć przez  $O(\log m)$  (a da się nawet uzyskać  $O(\log^* m)$ ), gdzie  $m$  jest ilością użyć operacji  $Make(v, D)$ . Dodatkowo struktura zapewnia możliwość sprawdzenia, czy zawiera więcej niż jeden zbiór (wystarczy pojedynczy licznik inkrementowany przy  $Make(v, D)$  i dekrementowany przy  $Union(u, v, D)$ ).

- strukturę  $T$  do przechowywania informacji o niektórych wierzchołkach jak binarne drzewo prefiksowe lub hashmapa, przechowywującą dla wierzchołka  $v_T$  informacje:
  - wskaźnik do wierzchołka  $v$  w strukturze  $D$
  - informacje o wadliwości/braku wadliwości wierzchołka
  - binarną informację o tym czy wierzchołek był odwiedzony i należy do  $F \cup N(F)$

wspierającą operacje:

- $Insert(v, T)$  wstawiającą wierzchołek  $v$  do struktury  $T$  i zwracającą wskaźnik na  $v_T$
- $Retrieve(v, T)$  zwracającą  $v_T$  lub  $NULL$  w przypadku gdy  $v$  nie ma w strukturze

które wymagają  $O(n)$  czasu na wykonanie.

Definiuję pomocniczą funkcję uzyskiwania wierzchołków ze struktury  $T$  i inicjalizowania w razie nieobecności:

```
Retrieve2(v, T){
  v_T = Retrieve(v, T);
  if(v_T == NULL){
    v_T = Insert(v, T);
    v_T.healthy = TRUE;
    v_T.visited = FALSE;
```

```

    Make( $v, D$ );
  }
  return( $v_T$ );
}

```

Najistotniejszą częścią algorytmu jest procedura (czasami dla wielu wierzchołków z  $F$ )  $DFS(f)$  znajdująca spójne składowe  $G(C) - C$ , dla  $C$  spójnej składowej  $Q_n^2[F]$  zawierającej wadliwy wierzchołek  $f$ , o następującym pseudokodzie:

```

DFS( $f$ ){
  foreach( $u \in N(f)$ ){
     $u_T = Retrieve2(u, T)$ ;
    if( $u_T.visited == FALSE$ ){
       $u_T.visited = TRUE$ ;
      if( $u_T.healthy$ ){
        foreach( $v \in N(u)$ ){
           $v_T = Retrieve2(v, T)$ ;
          if( $v_T.healthy$ ){
            if( $Find(u, D) \neq Find(v, D)$ )  \\ krawędź  $uv$  należy do  $G(C) - C$ 
          }else if( $v_T.visited == FALSE$ ){
             $v_T.visited = TRUE$ ;
            DFS( $v$ );  \\ wadliwy wierzchołek należący do  $C$ 
          }
        }
      }
    }else DFS( $u$ );  \\ wadliwy wierzchołek należący do  $C$ 
  }
}

```

Powyższa prodedura uruchamiana jest z funkcji głównej:

```

Conectivity( $n, F$ ){
   $T = empty\_structureT()$ ;
  foreach( $f \in F$ ){
     $f_T = Insert(f, T)$ ;
     $f_T.healthy = FALSE$ ;
     $f_T.visited = FALSE$ ;
  }
  foreach( $f \in F$ ){
     $f_T = Retrieve(f, T)$ ;
    if( $f_T.visited == FALSE$ ){
       $f_T.visited = TRUE$ ;
       $D = empty\_structureD()$ ;
      DFS( $f$ );
      if( $D.counter > 1$ ) return( $FALSE$ );
    }
  }
  return( $TRUE$ );
}

```

### 2.2.4. Analiza złożoności

**Wniosek 23.** *Algorytm ma pesymistyczną złożoność czasową i pamięciową  $O(|F| \cdot n^3)$ .*

*Dowód.* Dla każdego wierzchołka z  $F$  każdy sąsiad jest przeglądany po jeden raz, dla każdego wierzchołka należącego do  $N(F)$  również przeglądani są wszyscy sąsiedzi po razie. Przeglądnięcie jednego wierzchołka (znalezienie odpowiedniego wierzchołka w  $T$  i  $D$ ) zajmuje  $O(n)$ , ustawienie właściwości w  $D$  zajmuje stały czas po posiadaniu dowiązania do odpowiedniego wierzchołka – daje to złożoność tej części  $O(|F| \cdot n^3)$ .

Operacja  $Make(v, D)$  używana jest dla każdego wierzchołka z  $G(C) - C$  po razie dla każdego  $C$  (może być użyta więcej niż raz dla wierzchołków oddalonych o 2 od  $F$  i występujących w różnych  $G(C)$ ). W przypadku  $Find(v, D)$  i  $Union(u, v, D)$  uruchamiane są one maksymalnie odpowiednio dwa i jeden raz dla każdego z sąsiadów wierzchołków  $N(F)$  – daje to złożoność  $O(|F| \cdot n^2 \log(n))$ .

Preprocessing i Postprocessing (tworzenie i usuwanie struktur  $T$  i  $D$ ) może być zrobione w czasie liniowym od ich wielkości (w przypadku  $D$  i hashmapy można trzymać dodatkowo nieuporządkowaną listę dowiązań do wszystkich elementów). W przypadku struktury  $T$  wielkość tą można ograniczyć przez  $O(|F| \cdot n^3)$  przy użyciu drzewa prefikсового (lub  $O(|F| \cdot n^2)$  przy użyciu hashmapy, która nie pozwala jednak uzyskać odpowiedniej złożoności przy pesymistycznym scenariuszu), zaś w przypadku struktur  $D$  łącznie  $O(|F| \cdot n^2)$ .  $\square$

## 2.3. Wnioski i zastosowania

### 2.3.1. Cykl Eulera

Mając dostępny efektywny algorytm badania spójności wysnułem następujący wniosek:

**Twierdzenie 24.** *Dla  $F \subseteq V(Q_n)$  można rozstrzygnąć, czy w  $Q_n - F$  jest cykl Eulera w czasie  $O(|F| \cdot n^3)$*

*Dowód.* Kryterium istnienia cyklu Eulera jest to, że po pierwsze graf jest spójny, a po drugie z każdego wierzchołka wychodzi parzysta liczba krawędzi. Spójność można sprawdzić w czasie  $O(|F| \cdot n^3)$  przy pomocy algorytmu z poprzedniego podrozdziału. Wierzchołek nie mający wadliwego sąsiada ma stopień  $n$ , wystarczy więc policzyć tylko parzystość dla tych którzy takiego sąsiada mają. W czasie i pamięci  $O(|F| \cdot n^2)$  można wstawić wszystkich niewadliwych sąsiadów wierzchołków wadliwych do drzewa prefikсового zapamiętując w liściach krotność. Po wszystkim wystarczy dla  $2|n|$  sprawdzić czy wszystkie wstawione wierzchołki mają krotność parzystą, zaś dla  $2 \nmid n$  trzeba po pierwsze sprawdzić, że wszystkie wstawione wierzchołki mają krotność nieparzystą, a po drugie że jest ich dokładnie  $2^n - |F|$ .  $\square$

### 2.3.2. Długość ścieżki między dwoma punktami

**Fakt 25.** *Dla spójnego grafu  $Q_n - F$  pomiędzy każdymi dwoma wierzchołkami jest ścieżka długości nie większej niż  $n + n^2 \cdot |F|$ .*

*Dowód.* Jest to bezpośredni wniosek z dowodu lematu 22. Dwa dowolne wierzchołki są połączone w pełnej hiperkostce  $Q_n$  ścieżką długości co najwyżej  $n$ . W dowodzie zastępujemy (być może kilka razy) część takiej ścieżki inną ścieżką w grafie  $G(C) - C$ . Łącznie długość wszystkich takich zastąpień nie może wynosić więcej niż wynosi rozmiar  $G(F) - F$ , co dowodzi tezy.  $\square$

### 2.3.3. Istnienie ścieżki między dwoma punktami

Pomimo gorszej złożoności algorytmu ekspansywnego ma on wciąż przydatne zastosowania. Jeśli nie chcemy badać spójności całego grafu, a jedynie sprawdzić czy istnieje w nim ścieżka między dwoma wybranymi wierzchołkami algorytm ten łatwo zredukować:

**Twierdzenie 26.** *Dla  $F \subseteq V(Q_n)$  i dwóch wierzchołków  $u, v \in Q_n - F$  można w czasie  $O(n^{2.5}|F|)$  rozstrzygnąć, czy w  $Q_n - F$  istnieje  $uv$ -ścieżka.*

*Dowód.* Chcemy sprawdzić czy dwa wierzchołki należą do jednej spójnej składowej grafu  $Q_n - F$ . Zgodnie z lematem 9 jeśli spójne składowe w których znajdują się te wierzchołki są większe niż  $\frac{|F|}{\varepsilon}$ , to są składową główną (a więc tą samą), wystarczy więc z obu wierzchołków wystartować przeszukiwanie grafowe i zakończyć je jeśli zbada się tyle wierzchołków. Jeśli oba wyszukiwania zakończą się dzięki temu kryterium, lub jeśli zostanie napotkany ten drugi wierzchołek, to w  $Q_n - F$  istnieje  $uv$ -ścieżka, w przeciwnym przypadku nie.  $\square$



## Rozdział 3

# Długie ścieżki i cykle w grafie

W poprzednim rozdziale przedstawiony był przykład problemu na wadliwej hiperkostce, dla którego można było znaleźć rozwiązanie wielomianowe od  $n$  i  $|F|$ . W tym rozdziale przedstawimy kilka problemów, dla których samo przedstawienie wyników wymagało by wykładniczej pamięci, jednak samo rozstrzygnięcie czy rozwiązanie istnieje (sprawdzenie warunków twierdzenia) jest możliwe w czasie  $O(|F| \cdot n)$  dla odpowiednio małych  $|F|$  (wartości podane w sformułowaniach twierdzeń). W przypadku podwójnych ścieżek twierdzenie daje jedynie warunek wystarczający, dlatego algorytm otrzymany dzięki niemu nawet dla tych małych  $|F|$  potrafi jedynie rozstrzygnąć pomiędzy "istnieją długie ścieżki" i "kryterium nie rozstrzyga".

### 3.1. Definicje

**Definicja 3.1.1.** *Wolną od wad (nieprzechodzącą przez wierzchołki wadliwe) ścieżkę bez powtórzeń (drogę) w hiperkostce  $Q_n$  z wadami ze zbioru  $F \subseteq V(Q_n)$  nazwiemy długą, jeśli ma długość co najmniej  $2^n - 2|F| - 2$ .*

**Definicja 3.1.2.** *Wolny od wad cykl bez powtórzeń w hiperkostce  $Q_n$  z wadami ze zbioru  $F \subseteq V(Q_n)$  nazwiemy długim, jeśli ma długość co najmniej  $2^n - 2|F|$ .*

**Uwaga 27.** *Dla  $F \cup \{u, v\}$  należącego do jednej dwudzielnej części  $Q_n$  nie da się skonstruować  $uv$ -ścieżki wolnej od wad o długości większej niż  $2^n - 2|F| - 2$  (stąd długość w definicji).*

*Dowód.* Skoro  $Q_n$  jest dwudzielna, to każda ścieżka musi odwiedzić tyle samo wierzchołków w obu częściach (plus jeden koniec), ponieważ w części z  $u$  i  $v$  odwiedza co najwyżej  $2^{n-1} - |F|$ , to w drugiej co najwyżej  $2^{n-1} - |F| - 1$  - daje to długość  $2^n - 2|F| - 2$ .  $\square$

**Definicja 3.1.3.** *Wierzchołek  $v \in V(Q_n)$  jest otoczony przez  $F \subseteq V(Q_n)$  gdy  $N(v) \subseteq F$  ( $F$  zawiera wszystkich sąsiadów  $v$ ).*

**Definicja 3.1.4.** *Dla  $u, v \in V(Q_n)$ ,  $F \subseteq V(Q_n)$  trójka  $(u, v, F)$  jest zablokowana w  $Q_n$  gdy  $u$  jest otoczony przez  $\{v\} \cup F$  lub  $v$  jest otoczony przez  $\{u\} \cup F$ .*

### 3.2. Długie ścieżki

**Twierdzenie 28.** *Dla  $Q_n$  i  $F \subseteq V(Q_n)$ , takich, że  $2 \leq n \leq 5$  i  $|F| \leq 2n - 4$  dla  $u, v \in V(Q_n) \setminus F$  takich, że trójka  $(u, v, F)$  nie jest zablokowana w  $Q_n$  długa  $uv$ -ścieżka bez wad nie istnieje tylko wtedy, gdy  $n = 4$  oraz istnieją takie  $a, b \in V(Q_n)$ , że  $d(a, b) = 4$  i  $F \cup \{u, v, a, b\}$  jest dwudzielną częścią  $Q_n$ .*

*Dowód.* Łatwo rozpatrzyć wszystkie przypadki. □

**Twierdzenie 29.** Dla  $Q_n$  i  $F \subseteq V(Q_n)$ , takich że  $n \geq 6$  i  $|F| \leq 2n - 4$  dla każdych  $u, v \in V(Q_n) \setminus F$  jeśli  $(u, v, F)$  nie jest zablokowane w  $Q_n$ , to istnieje długa  $uv$ -ścieżka bez wad.

*Dowód.* (krótki szkic dowodu z pracy [4])

Dowód oparty jest na indukcji po wymiarze. Podstawę indukcji stanowi twierdzenie 28. Dla  $n \geq 6$ ,  $|F| \leq 2n - 4$  można podzielić  $Q_n$  na dwie kostki  $Q_{n-1}$  wybierając jedną z  $n$  współrzędnych i definiując podkostki  $Q_{n-1}^0$  i  $Q_{n-1}^1$  jako rozpięte przez wierzchołki mające na tej współrzędnej odpowiednio 0 i 1. Dla  $|F| \leq 2n - 5$  łatwo jest dobrać współrzędną tak, żeby każda z podkostek miała co najwyżej  $2n - 6 = 2(n - 1) - 4$  wadliwych wierzchołków (wystarczy wybrać dowolne  $f_1, f_2 \in F$  i podzielić według jednej ze współrzędnych różniących ich ciągi binarne). Dla  $|F| = 2n - 4$  można rozpatrzyć macierz  $|F| \times n$ , w której w wierszach wypisane są ciągi binarne wszystkich wierzchołków wadliwych. Trzeba wybrać taką kolumnę, w której zarówno 0 jak i 1 jest co najmniej po 2. Gdyby nie dało się dokonać takiego wyboru oznaczałoby to, że w każdej kolumnie jest albo co najwyżej jedno 0 albo co najwyżej jedna 1, przez proste zanegowanie jednej współrzędnej w całej kostce (ta operacja nie zmienia nic poza numerowaniem) można uzyskać przypadek, że w każdej kolumnie jest co najwyżej jedna 1. Ponieważ kolumn jest tylko  $n$ , zaś każda zawiera co najwyżej jedną 1, to oznaczałoby to, że może mieć tylko  $n + 1$  różnych wierszy  $\Rightarrow 2n - 4 = |F| \leq n + 1 \Rightarrow n \leq 5$  (a więc ponieważ  $n \geq 6$ , to zawsze istnieje wybór współrzędnej).

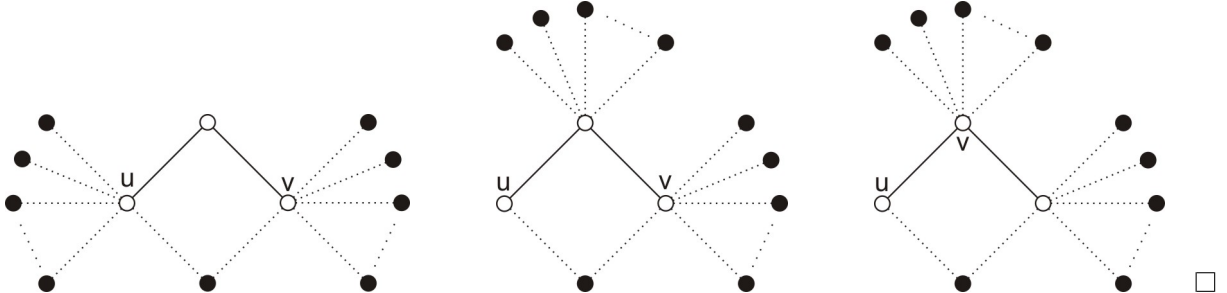
Dalej przy użyciu lematów:

- Dla  $|F| \leq 2n - 3$  co najwyżej jeden z wierzchołków jest otoczony przez  $F$ .
- Dla  $|F| \leq 2n - 4$  i ustalonego nieotoczonego wierzchołka  $u$  istnieje co najwyżej jeden wierzchołek  $v$  taki, że  $(u, v, F)$  jest zablokowana.
- Dla  $|F| \leq 2n - 5$  tylko jedna trójka  $(u, v, F)$  może być zablokowana, i to taka, że  $uv \in E(Q_n - F)$ .

i wykorzystując fakt, że w podkostkach poza kilkoma przypadkami istnieją odpowiednie długie kostki rozważa się dużą liczbę przypadków (rozbiecie ze względu na należenie  $u$  i  $v$  do tej samej/różnej podkostki, bycia otoczonym/zablokowanym/wolnym w podkostce). Dla każdego z tych przypadków da się pokazać metodę łączenia długich ścieżek z podkostek. □

**Uwaga 30.** Dla  $Q_n$ ,  $F \subseteq V(Q_n)$ ,  $|F| = 2n - 3$  teza twierdzenia 29 przestaje być prawdziwa.

*Dowód.* Dla każdego  $n \geq 4$  istnieje po kilka przypadków w których  $|F| = 2n - 3$ ,  $(u, v, F)$  nie jest zablokowane, ale nie ma długiej  $uv$ -ścieżki bez wad. 3 przykłady :



### 3.3. Długie cykle

**Definicja 3.3.1.** Dla zbioru  $D \subseteq [n]$ ,  $d = |D|$  oraz  $u \in \{0, 1\}^{n-d}$  definiujemy kostkę  $Q_D(u)$  jako  $d$  wymiarową podkostkę  $Q_n$ , której współrzędne spoza  $D$  są ustalone przez wektor  $u$ . Definiujemy również  $V_D(u) = \{(u, v)_D : v \in \{0, 1\}^d\}$  (wierzchołki z oryginalnej kostki wzięte do  $Q_D(u)$ ), oraz  $F_D(u) = F \cap V_D(u)$ .

**Lemat 31.** Niech  $F \subseteq V(Q_n)$  takie, że  $|F| \geq 2n$  i niech  $d = \lceil \frac{n^2}{2|F|-n-2} \rceil$ . Wtedy istnieje zbiór  $D \subseteq [n]$ ,  $|D| = d$ , taki że  $|F_D(u)| \leq d + 1$  dla każdego  $u \in \{0, 1\}^{n-d}$ .

Lemat pochodzi z pracy [5] i został zmodyfikowany do tej postaci w pracy [4] aby lepiej pasować do dowodu poniższego twierdzenia.

**Twierdzenie 32.** Dla  $n \geq 15$  i  $F \subseteq V(Q_n)$ , takiego że  $|F| \leq \frac{n^2}{10} + \frac{n}{2} + 1$  istnieje długi cykl bez wad.

*Dowód.* (krótki szkic dowodu z pracy [4])

Na podstawie lematu 31 znajdujemy zbiór  $D \subseteq [n]$ , taki że  $|F_D(u)| \leq 2d - 4$  dla każdego  $u \in \{0, 1\}^{n-d}$ . Dla dowolnego cyklu Hamiltona  $(u_0, u_1, \dots, u_{2^n-d} = u_0)$  w  $Q_{n-d}$  dobieramy w kostce  $Q_D u_i$  dwa nie wadliwe wierzchołki  $a_i$  oraz  $b_i$ , takie że  $a_i b_{i+1} \in E(Q_n)$  dla każdego  $i \in [2^{n-d}]$  (modulo  $2^{n-d}$ ), oraz  $(a_i, b_i, F_D(u^i))$  nie zablokowane (choć jest to nietrywialne to da się takie dobrać). Na podstawie twierdzenia 29 wierzchołki  $a_i$  i  $b_i$  są łączone długimi ścieżkami dając cykl długości  $\geq 2^n - 2|F|$ . Ograniczenie  $|F| \leq \frac{n^2}{10} + \frac{n}{2} + 1$  potrzebne jest po to, aby  $\lceil \frac{n^2}{2|F|-n-2} \rceil \geq 5$  omijając złe przypadki z twierdzenia 28.  $\square$

### 3.4. Długie pary ścieżek

**Lemat 33.** Dla  $n \geq 2$ ,  $F \subseteq V(Q_n)$ ,  $|F| \leq n - 2$  dla każdych dwóch  $u, v \in V(Q_n - F)$  istnieje długa  $uv$ -ścieżka bez wad.

*Dowód.* Bezpośrednio z 29, gdzie ze względu na rozmiar  $F$  trójka  $(u, v, F)$  nie może być zablokowana.  $\square$

**Twierdzenie 34.** Dla  $F \subseteq V(Q_n)$ ,  $|F| \leq n - 3$  niech  $A$  i  $B$  będą różnymi dwuelementowymi podzbiorami  $V(Q_n) - F$ , takimi że  $A \cup B$  nie należy do jednej części dwudzielnej kostki. Wtedy istnieje para wierzchołkowo rozłącznych ścieżek o łącznej długości  $\geq 2^n - 2|F| - 3$  zaczynających się w wierzchołku z  $A$  i kończących na wierzchołku z  $B$ .

*Dowód.* (krótki szkic dowodu z pracy [6])

Dowód podobnie jak inne przebiega indukcyjnie – małe przypadki ( $n \leq 5$ ) można sprawdzić ręcznie (tutaj trochę więcej sprawdzania niż w poprzednich dowodach), dla większych łatwo jest podzielić  $Q_n$  na dwie  $Q_{n-1}$  tak, żeby każda z nich miała nie więcej niż  $n - 4$  wierzchołków wadliwych. Dalej rozpatrywane jest dużo przypadków w zależności od podziału wierzchołków z  $A$  i  $B$  na dwie podkostki i w każdym z możliwych przypadków łączy się podwójne i pojedyncze ścieżki istniejące na mocy indukcji i twierdzenia 29.  $\square$

**Uwaga 35.** Jeśli  $A = \{u, w\}$ ,  $B = \{v, w\}$ , to jedna ze ścieżek musi mieć długość 0 i być zaczepiona w wierzchołku  $w$ . Twierdzenie 34 daje wtedy  $uv$ -ścieżkę wolną od wad długości  $2^n - 2|F| - 3 = 2^n - 2|F \cup \{w\}| - 1$ , a więc o jeden dłuższą niż w twierdzeniu 29 (możliwe jest to tylko dlatego, że  $u$  i  $v$  nie należą do jednej części dwudzielnej  $Q_n$ ).



## Rozdział 4

# Ścieżka i cykl Hamiltona

### 4.1. Kostka bez wadliwych wierzchołków

Pełna kostka  $Q_n$  ( $n \geq 2$ ) jest grafem hamiltonowskim (posiadającym cykl Hamiltona) – bardzo wiele różnych cykli Hamiltona można uzyskać już łącząc dwa cykle z podkostek  $Q_{n-1}$  (wystarczy, że w dwóch takich cyklach jest taka sama krawędź). Cykle hamiltona na hiperkostce mają również zastosowania praktyczne. Jednym z przykładów użycia takich cykli są kody Graya – ciągi wektorów binarnych określonej długości, które zawierają wszystkie takie wektory, oraz posiadają tę własność, że dwa sąsiednie wektory różnią się tylko na jednej pozycji. Kody te znajdują zastosowanie między innymi przy minimalizacji kodowania i obwodów binarnych, oraz w algorytmach genetycznych i kodach korekcyjnych.

Pojawia się pytanie, czy po usunięciu kilku wierzchołków nadal da się zbudować takie cykle lub chociaż ścieżki Hamiltona.

### 4.2. Ogólne podejście do problemu

W przypadku ogólnym sprawdzenie, czy graf posiada cykl Hamiltona jest problemem NP-zupełnym, dlatego przy dowolnym wyborze zbioru wierzchołków usuniętych ciężko spodziewać się algorytmu w czasie  $O(2^n)$ , a co dopiero w  $O(n^c)$ . Jeśli jednak zbiór usuniętych wierzchołków jest odpowiednio mały lub tworzy graf z jakiejś wąskiej klasy, to istnieją dowody istnienia takiego cyklu.

**Definicja 4.2.1.** *Graf dwudzielny nazywamy zbalansowanym jeśli jego dwie części dwudzielne mają ten sam rozmiar, zaś prawie zbalansowanym jeśli ich rozmiary różnią się o 1.*

Hiperkostka jest grafem dwudzielnym, więc i po usunięciu części wierzchołków takim pozostanie, jako że dowolny cykl w takim grafie odwiedza tyle samo wierzchołków w obu częściach dwudzielnych (z krotnościami), to podstawowym warunkiem koniecznym jest to, żeby z obu tych części usunąć dokładnie tyle samo wierzchołków. W przypadku ścieżki Hamiltona koniecznym jest żeby graf był zbalansowany lub prawie zbalansowany.

**Definicja 4.2.2.** *Graf dwudzielny (z częściami  $U$  i  $v$ ) nazywamy hamiltonowsko wiązalnym jeśli:*

- *jest zbalansowany i dla każdej pary  $u \in U, v \in V$  istnieje ścieżka Hamiltona z  $u$  do  $v$ .*
- *jest prawie zbalansowany i dla każdej pary  $u, u' \in U$  (gdzie  $U$  to ta większa składowa) istniała ścieżka Hamiltona z  $u$  do  $v$ .*

Jeśli graf jest zbalansowany i hamiltonowsko wiązalny, to istnieje w nim cykl Hamiltona (jeśli ma więcej niż 2 wierzchołki).

Hiperkostka jest grafem hamiltonowsko wiązalnym.

W 2001 Stephen Locke postawił hipotezę:

**Przypuszczenie 1.** *Niech  $F \subseteq V(Q_n)$  będzie zbiorem po  $k$  wierzchołków z obu dwudzielnych składowych  $Q_n$ . Wtedy  $Q_n - F$  posiada cykl hamiltona jeśli  $n \geq k + 2$ .*

Od tego czasu udowodnione zostały szczególne przypadki tej hipotezy ( $k \leq 3$ ,  $k \leq \frac{n-5}{6}, \dots$ ), a w 2009 podane zostało ich uzupełnienie – dowód indukcyjny po  $n$  w pracy [7], który jednak po kilka przypadków granicznych odsyła do drugiej pracy, którą autorzy mają dopiero opublikować.

### 4.3. Dowody dla szczególnych przypadków F

**Definicja 4.3.1.** *Podgraf  $H$  grafu  $G$  nazywamy izometrycznym jeśli  $d_H(u, v) = d_G(u, v)$  dla każdych  $u, v \in V(H)$ .*

Łatwo zauważyć, że podgraf izometryczny jest zawsze podgrafem indukowanym przez podzbiór wierzchołków, implikacja w drugą stronę jednak nie zachodzi (na przykład  $P_4$  jako podgraf  $C_5$ ).

#### 4.3.1. podkostka

**Fakt 36.** *Jeśli jako  $F \subseteq V(Q_n)$ ,  $n \geq 3$  weźmiemy zbiór wierzchołków podkostki  $Q_m$  ( $1 \leq m < n$ ), to  $Q_n - F$  posiada cykl Hamiltona.*

*Dowód.* Dla  $m \geq 2$ :

Porządkany cykl Hamiltona w  $Q_n - F$  można otrzymać poprzez podział grafu na  $2^{n-m} - 1$  kostek  $Q_m$ , wybranie tego samego cyklu Hamiltona dla tych podkostek a następnie połączenie ich w następujący sposób:

Wybieramy jedną z podkostek, która wraz z usuniętą tworzy kostkę  $Q_{m+1}$  – taka kostka różni się od poprzedniej na dokładnie jednej współrzędnej (takiej która jest stała na całej podkostce). Łączymy obie i otrzymujemy w ten sposób wadliwą kostkę  $Q_{m+1}$  z cyklem Hamiltona. Pozostałe kostki łączymy parami, razem z cyklami (wystarczy wybrać taką samą krawędź w obu i dokonać "przepięcia") – wszystkie tak samo. Cykl z wadliwej hiperkostki ma tylko jedną krawędź, której nie ma w innych cyklach – można dokonać połączenia wadliwej  $Q_{m+1}$  z całą i dalej aż do otrzymania  $Q_n - F$  z cyklem Hamiltona.

Dla  $m = 1$  postępujemy analogicznie, jednak po pierwszym kroku otrzymana  $Q_2 - Q_1$  nie ma cyklu a jedynie ścieżkę Hamiltona, jednak dalej  $Q_3 - Q_1$  posiada już cykl Hamiltona, który można przedłużyć na cykl Hamiltona na  $Q_3$  zawierający tę krawędź z  $Q_2 - Q_1$ . Dalej można postępować już tak samo – zawsze da się wybrać krawędź według której można dokonać "przepięcia".  $\square$

Podobnym sposobem można otrzymać cykle dla  $F$  równemu sumie kilku kostek ( $m \geq 2$ ), które nie są połączone (tak długo, jak jest ich na tyle mało, aby dało się wybierać krawędzie do "przepięcia" – to da się sprawdzić algorytmem wielomianowym od  $n$  i  $F$ ).

Poniżej przedstawię szkice dowodów twierdzeń dla innych szczególnych przypadków  $F$  opisanych w pracy [8]. Ze względu na licznosc mało istotnie różniących się przypadków po

obrazki i dokładne opisy wyborów wierzchołków, które łączymy ścieżkami Hamiltona w podkostkach odsyłam to tejsze pracy.

### 4.3.2. Ścieżka izometryczna

**Fakt 37.** *W ścieżce izometrycznej w  $Q_n$  każda z krawędzi ma inny kierunek, a więc ścieżka taka może mieć długość co najwyżej  $n$ .*

*Dowód.* Gdyby dwie krawędzie miały ten sam kierunek, to wierzchołki z początku pierwszej i końca drugiej można w  $Q_n$  połączyć ścieżką o 2 (przesunięta ścieżka pomiędzy tych dwóch krawędzi).  $\square$

**Twierdzenie 38.** *Dla  $F \subseteq V(Q_n)$  indukującego ścieżkę izometryczną nieparzystej długości  $Q_n - F$  jest hamiltonowsko wiązalna jeśli  $n \geq 4$ .*

*Dowód.* Przypadek ścieżki długości 1 indukcyjnie po  $n$  – dzielimy kostkę na dwie w kierunku  $i$  – innym niż ten krawędzi ścieżki usuniętej. Jedna podkostka jest pełna, zaś w drugiej jest usunięta ścieżka. Jeśli wierzchołki  $u$  i  $v$  które chcemy połączyć należą do tej samej podkostki, to łączymy je ścieżką Hamiltona w tej części na podstawie indukcji. Wybieramy dwa kolejne wierzchołki na tej ścieżce, nie będące w  $Q_n$  sąsiadami  $F$  w kierunku  $i$ , i łączymy w drugiej części ścieżką Hamiltona ich sąsiadów w tym kierunku.

Jeśli dwa wierzchołki są w różnych częściach, wystarczy wybrać dowolną parę wierzchołków, takich że: są sąsiadami w kierunku  $i$ , są różne od  $u$  i  $v$ , oraz należą do innych dwuspójnych składowych niż wierzchołek  $u$  lub  $v$  z ich części – można wtedy znaleźć ścieżki Hamiltona w obu częściach pomiędzy wierzchołkami  $u$  i  $v$  oraz tymi wybranymi i je połączyć.

Dla ścieżki większej długości dzielimy hiperkostkę na dwie względem kierunku drugiej krawędzi na ścieżce. W każdej podkostce znajduje się część ścieżki, która jest izometryczną nieparzystą ścieżką. Ścieżkę Hamiltona uzyskujemy jak poprzednio – przy rozbiciu na dwa przypadki łączenia ścieżek z podkostek. Przypadek  $n = 4$  jest natomiast rozważany ręcznie (tam są możliwe tylko 2 takie ścieżki z dokładnością do zmiany wierzchołka początkowego i permutacji współrzędnych).  $\square$

**Twierdzenie 39.** *Dla  $F \subseteq V(Q_n)$  indukującego ścieżkę izometryczną parzystej długości ( $\geq 2$ )  $Q_n - F$  jest hamiltonowsko wiązalna jeśli  $n \geq 5$ .*

*Dowód.* Ponownie indukcyjnie po  $n$ , tym razem dzieląc kostkę w kierunku pierwszej krawędzi ścieżki – zostawiając w pierwszej podkostce pojedynczy wadliwy wierzchołek, zaś w drugiej kostce wymiaru  $\geq 4$  wadliwą ścieżkę nieparzystej długości (przypadek z poprzedniego twierdzenia). Tym razem należy rozpatrzyć 3 przypadki (gdy  $u$  i  $u'$  są w różnych kostkach i 2 przypadki, gdy są w tej samej). Dla każdego z tych przypadków ponownie łatwo wybrać odpowiednie ścieżki w podkostkach i połączyć je tak aby uzyskać ścieżki Hamiltonowskie w  $Q_n - F$  od  $u$  do  $u'$  (szczegóły z obrazkami można zobaczyć w pracy [8]).  $\square$

**Uwaga 40.** *Na podstawie [9] prawdziwe jest mocniejsze stwierdzenie:*

*Niech  $P$  będzie ścieżką w długości  $\leq 2n - 4$  w grafie  $Q_n$ , wtedy  $Q_n - P$  jest hamiltonowsko wiązalna.*

### 4.3.3. Cykl izometryczny

**Uwaga 41.** W cyklu izometrycznym na hiperkostce każdy kierunek krawędzi występuje dokładnie 2 razy, w antypodycznych krawędziach cyklu.

*Dowód.* Każdy kierunek krawędzi musi się pojawić na dowolnym cyklu w hiperkostce parzystą ilość razy. Dla cyklu długości  $2k$  weźmy dwa wystąpienia tego samego kierunku, oddalone od siebie na cyklu o minimalną liczbę krawędzi. Dostajemy ścieżkę długości  $l$  zaczynającą i kończącą się krawędzią tego samego kierunku. Podobnie jak w dowodzie faktu 37 mamy w hiperkostce inną ścieżkę długości  $l - 2$  łączącą te same końce. Z izometryczności  $l - 2 \geq \min(l, 2k - l)$ , zaś z minimalności  $l - 2 \leq 2k - l \Rightarrow l - 1 = k$ .  $\square$

**Twierdzenie 42.** Niech  $C$  będzie izometrycznym cyklem w  $Q_n$  ( $n \geq 6$ ), o długości  $2k$  podzielnej przez 4, wtedy  $Q_n - P$  jest hamiltonowsko wiązalna.

*Dowód.* Zaczynamy od podzielenia hiperkostki na dwie w kierunku wyznaczonym przez pierwszą (dowolnie wybraną) krawędź cyklu. W obu kostkach znajduje się wadliwa izometryczna ścieżka długości nieparzystej, a więc z twierdzenia 38 obie są hamiltonowsko wiązalne. Podobnie jak wcześniej wystarczy rozważyć dwa przypadki. Jeśli dwa wierzchołki  $u$  i  $v$  które chcemy połączyć ścieżką Hamiltona należą do różnych podkostek, to wystarczy dobrać po jednym wierzchołku z podkostek, tak żeby te wierzchołki były sąsiadami, były różne od  $u$  i  $v$  i należały do odpowiednich części dwuspójnych – tworzymy ścieżki w podkostkach i łączymy je ze sobą. Jeśli zaś  $u$  i  $v$  należą do tej samej podkostki, to wystarczy na ścieżce Hamiltona pomiędzy nimi wybrać dowolną krawędź, taką że sąsiedzi jej końców z drugiej hiperkostki nie są wadliwi – pomiędzy tymi sąsiadami wyznaczamy ścieżkę w drugiej podkostce i ”dopinamy” do ścieżki z pierwszej podkostki.  $\square$

### 4.3.4. Drzewo izometryczne

**Uwaga 43.** W drzewie izometrycznym w  $Q_n$  każda z krawędzi ma inny kierunek.

*Dowód.* W przeciwnym przypadku istnieją dwie krawędzie o tym samym kierunku – w drzewie jest ścieżka, która zawiera obie krawędzie, co daje sprzeczność z lematem 37 (w drzewie każde dwa wierzchołki są połączone tylko jedną ścieżką, więc nie ma innej krótszej).  $\square$

**Uwaga 44.** W niepustym drzewie zbalansowanym, lub prawie zbalansowanym z przewagą w części dwudzielnej  $V$  jest liść zawarty w tej składowej.

*Dowód.* Nie wprost – jeśli w drzewie każdy liść należy do dwudzielnej części  $U$ , to można wybrać dowolny wierzchołek z  $v \in V$  jako korzeń. Usunięcie dowolnej gałęzi bez rozwidleń (od ostatniego rozwidlenia do liścia) nie produkuje nowych liści, może też pogorszyć balans jedynie zwiększając przewagę  $V$ . Gdy zostanie już tylko jedna ścieżka, to kończyć się ona będzie z obu stron wierzchołkami z  $U$ , co daje sprzeczność.  $\square$

**Lemat 45.** Dla zbalansowanego drzewa  $T$  izometrycznego w  $Q_5$  graf  $Q_5 - T$  jest hamiltonowsko wiązalny.

Lemat można sprawdzić ręcznie lub komputerowo (są tylko dwa takie drzewa nie będące ścieżkami z dokładnością do umiejscowienia w kostce).

**Twierdzenie 46.** Dla  $T$  zbalansowanego lub prawie zbalansowanego drzewa izometrycznego w  $Q_n$  graf  $Q_n - T$  jest hamiltonowsko wiązalny dla  $n \geq 5$  (w przypadku zbalansowanego wystarczy  $n \geq 4$ ).



*Dowód.* Dowód przez indukcję po rozmiarze hiperkostki. Jako krok bazowy można traktować lemat 45, przypadek drzewa prawie zbalansowanego (w  $Q_5$  istnieje takie tylko jedno nie będące ścieżką) i zbalansowanego dla  $n = 4$  (tu są tylko ścieżki) można rozpatrzyć analogicznie dla tamtego przypadku. W kroku indukcyjnym dla drzewa zbalansowanego korzystamy z wymiaru o 1 mniejszego i drzewa prawie zbalansowanego zaś dla drzewa prawie zbalansowanego korzystamy z poprzedniego kroku dla drzewa zbalansowanego. Dla drzewa prawie zbalansowanego na podstawie uwagi 44 wybieramy liść z części dwudzielnej w której ma ono więcej wierzchołków i dzielimy hiperkostkę na dwie mniejsze według kierunku krawędzi drzewa wchodzącej do tego wierzchołka. Otrzymujemy w ten sposób w jednej podkostce pojedynczy wadliwy wierzchołek (jak w dowodach poprzednich twierdzeń) oraz w drugiej drzewo izometryczne zbalansowane. Jak we wcześniejszych dowodach w zależności od tego w których podkostkach znajdują się wierzchołki które chcemy połączyć ścieżką Hamiltona tworzymy ścieżki w podkostkach i łączymy je. Dla drzewa zbalansowanego postępujemy analogicznie z tą różnicą, że wybór liścia jest dowolny.  $\square$



## Rozdział 5

# Największa odległość grafowa

**Definicja 5.0.2.** Średnicą grafu nazywamy największą odległość pomiędzy jego dwoma wierzchołkami  $d(G) = \max_{u,v \in V(G)} d(u,v)$ .

Jedną z najważniejszych zalet hiperkostki  $Q_n$ , jest to, że pomimo względnie niskiej gęstości jej średnica jest niewielka i równa  $n$ , a do tego po usunięciu części węzłów często pozostaje spójna. Graf ten posiada również tę przyjemną własność, że przy losowym usuwaniu wierzchołków odległości pomiędzy nieusuniętymi wierzchołkami rzadko się zwiększają (w szczególności dla wierzchołków bardziej oddalonych). Przy intencjonalnym usuwaniu aby zwiększyć odległość pomiędzy dowolnymi dwoma wierzchołkami potrzeba usunąć co najmniej dwa wierzchołki (w przypadku dowolnych wierzchołków oddalonych o 2), zaś aby zwiększyć średnicę grafu należy usunąć co najmniej  $n - 1$  wierzchołków (wszystkich prócz jednego sąsiadów dowolnie wybranego wierzchołka). Aby uzyskać średnicę  $n + 2$  należy usunąć  $2n - 3$  wierzchołków (na przykład sąsiedzi wierzchołków numer 1 i 2 oprócz 4 i siebie nawzajem przy numerowaniu klasycznym), dalej jednak pojawia się coraz więcej możliwości usuwania wierzchołków i ciężko wybrać najlepszą.

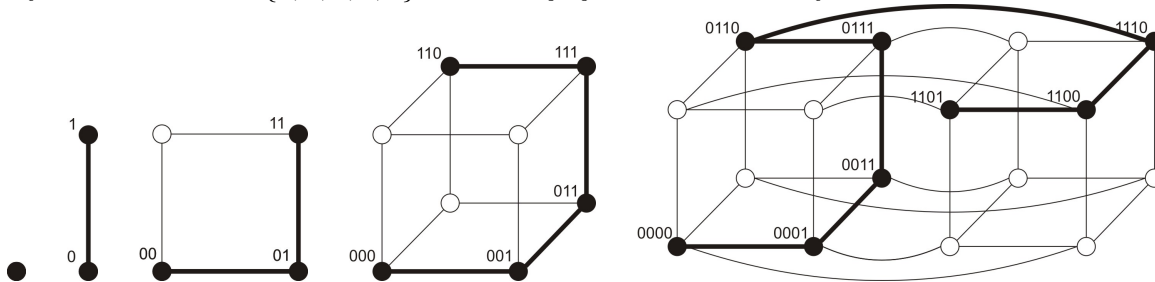
Zainspirowany tym postawiłem sobie pytanie jaką największą średnicę można uzyskać w ten sposób pozostawiając graf spójnym. Średnicę spójnego grafu  $Q_n - F$  wyznacza najkrótsza ścieżka między pewnymi dwoma wierzchołkami, jeśli więc dla jakiegoś grafu uda się uzyskać taką maksymalną średnicę, to można wybrać tę ścieżkę i odrzucić wszystkie wierzchołki spoza niej (nie zmniejszając średnicy). Oznacza, to że to maksimum po wszystkich grafach  $Q_n - F$  jest uzyskiwane dla grafu będącego ścieżką. Ścieżka taka ma tę własność, że ciągi binarne sąsiadujących w niej wierzchołków różnią się na dokładnie jednej pozycji, natomiast dla dowolnych dwóch nie będących sąsiadami ciągi te muszą się różnić na co najmniej dwóch pozycjach. Wyjściowy problem został więc zredukowany (okazał się równoważny) do problemu:

Znajdź najdłuższy ciąg wektorów binarnych długości  $n$ , że każde dwa sąsiednie wektory różnią się na co najmniej jednej pozycji, zaś każde dwa inne na co najmniej dwóch.

### 5.1. Ścieżka Fibonacciego

**Twierdzenie 47.** Dla danego  $n$  istnieje prosty algorytm konstruujący ciąg wektorów reprezentujący ścieżkę długości  $F_{n+1}$ , (gdzie  $F_n$  to  $n$ -ty wyraz ciągu Fibonacciego:  $F_0 = F_1 = 1$ ), która jest podgrafem indukowanym hiperkostki. Co więcej prefiks długości  $F_n$  jest takim ciągiem dla  $n - 1$  (po odrzuceniu ostatniej współrzędnej wektorów).

*Dowód.* Na potrzeby dowodu nazwę taką ścieżkę  $P_n$ . Dowód tradycyjnie przez indukcję po wymiarze – dla  $n \in \{0, 1, 2, 3, 4\}$  konstrukcja jest widoczna na rysunkach:



Dla tak małych wymiarów jest to jedyny z dokładnością do wyboru punktu początkowego i przepermutowania współrzędnych sposób uzyskania ścieżki tej długości. Już na tych przypadkach widać, że ścieżka tak powstała wygląda identycznie od początku i od końca (przy zamianie punktów początkowego i końcowego oraz przepermutowaniu współrzędnych). Dodatkowo widać też, że tylko 3 podkostki  $Q_{n-2}$  zawierają wybrane do ścieżki wierzchołki, dokładniej pierwsza podkostka zawiera  $P_{n-2}$ , druga  $P_{n-3}$  i trzecia znów  $P_{n-2}$  (znów "z dokładnością").

Konstrukcja indukcyjna  $P_n$  przebiega następująco:

Startuję od  $P_{n-1}$ , który nie używa tylko współrzędne od 0 do  $n-2$ , kolejnym wierzchołkiem w ciągu będzie sąsiad aktualnie ostatniego w kierunku  $n-1$ . Mam dane już ponad pół kostki, jeśli więc chcę uzyskać  $P_n$  takie samo od początku jak i od końca wystarczy znaleźć odpowiednią permutację. Co do tej nowej części mamy pewną dowolność przy wyborze permutacji, jednak musi się ona zgadzać na tej środkowej części ( $P_{n-2}$  zaczynające się po krawędzi w kierunku  $n-3$ , a kończące przed dostawioną  $n-2$  – te współrzędne zostaną ze sobą zamienione w permutacji i w całej ścieżce  $P_n$  krawędzie w tych kierunkach występują tylko po razie). Z kroku indukcyjnego  $P_{n-2}$  jest symetryczne w ten sposób, więc istnieje odpowiednia permutacja na kierunkach w niej używanych, wystarczy więc ją rozszerzyć o tą zamianę  $n-1$  z  $n-2$  (i formalnie dowolnie uzupełnić o kierunki w niej nie występujące, choć tak na prawdę takich nie ma). Z konstrukcji wynika, że zbudowana tak ścieżka posiada własność: dowolne dwa wierzchołki ze ścieżki są sąsiadami w hiperkostce tylko wtedy gdy są sąsiadami na ścieżce wewnątrz trzech swoich części. Pomiędzy pierwszą i drugą częścią własność ta jest spełniona z kroku indukcyjnego, zaś pomiędzy trzecią i drugą jest tak samo z symetrii. Pozostają części pierwsza i trzecia, ale skoro są one całkowicie zanurzone w dwóch podkostkach, które nie mają między sobą krawędzi, to warunek ten jest również spełniony w tym przypadku.  $\square$

**Wniosek 48.** *Jako, że ciąg Fibonacciego rośnie w tempie wykładniczym – ilość użytych wierzchołków wynosi  $\frac{1}{\sqrt{5}}((\frac{1+\sqrt{5}}{2})^{n+1} - (\frac{1-\sqrt{5}}{2})^{n+1}) \approx \frac{(1.62)^{n+1}}{\sqrt{5}} = \Omega(1.61^n)$ , przy wielkości kostki  $2^n$ .*

## 5.2. Wyniki eksperymentalne

**Uwaga 49.** *Dla wymiarów większych niż 4 da się otrzymać ścieżki jeszcze dłuższe – dla  $n = 5$  najdłuższa możliwa ścieżka jest:*

*00000 00001 00011 00111 01111 01110 01100 11100 11101 11001 11011 11010 10010 10110.*  
*Ma ona długość 13, jest więc o 1 dłuższa od ścieżki otrzymanej na podstawie twierdzenia 47*

Aby znaleźć możliwie najdłuższe ścieżki dla większych  $n$  przeprowadziłem eksperymenty przy pomocy komputera. Wyszukiwanie ścieżek prowadziłem startując od wierzchołka

$(0, \dots, 0)$  i dobierając do istniejącej ścieżki nowy wierzchołek, który jest sąsiadem aktualnego końca i nie jest sąsiadem żadnego poprzedniego, aż do momentu, gdy taki wybór nie jest możliwy – w ten sposób otrzymuje "grę". Znalezienie wszystkich możliwych ścieżek otrzymywalnych w ten sposób (a są to wszystkie takie poprawne ścieżki, które zaczynają się w wierzchołku  $(0, \dots, 0)$ ) otrzymywane jest poprzez przejście drzewa gry, której wierzchołkami są ścieżki, zaś ścieżki w przodkach tego wierzchołka są jej prefiksami. Drzewo tej gry ma maksymalny stopień rozgałęzienia równy  $n$ , zaś głębokość  $|P|$ .

Daje to złożoność algorytmu przeglądającego wszystkie ścieżki mocno pesymistycznie ograniczoną przez  $O(n^{|P|})$ . Nawet po zastosowaniu różnych usprawnień, algorytm dokładny szybko staje się zbyt wolny, dlatego dla  $n \geq 7$  musiałem skorzystać z metod losowych.

### 5.2.1. NMCS

Aby otrzymać pewną ścieżkę, która posiada tę własność, że nie da się jej już wydłużyć do innej poprawnej można przeprowadzić zejście po drzewie od korzenia do liścia poprzez losowe wybieranie jednego z dzieci za każdym razem, gdy taki wybór jest możliwy.

Przeprowadzając odpowiednio dużo takich losowych zejść po drzewie i wybierając najdłuższą ścieżkę ze znalezionych możemy uzyskać już w miarę dobry wynik, istnieją jednak metody, które pozwalają na lepsze ukierunkowanie losowości.

Jedną z takich metod jest przeszukiwanie Monte Carlo polegające na tym, że mając dany stan gry (aktualną ścieżkę) dla każdego możliwego ruchu (wyboru kolejnego wierzchołka) szacujemy jego wartość poprzez uruchomienie przeszukiwania losowego w tym kierunku i następnie wybranie najlepszego z nich. Metoda ta jest wykorzystywana głównie w przeypadkach gier w tym takich, w których gracz ma tylko częściowy wpływ na rozgrywkę (istnieje przeciwnik), jednak w przypadku szukania możliwie długiej ścieżki również się sprawdza.

Rozszerzenie tej metody zostało zaprezentowane w pracy [10] zwane Nested Monte Carlo Search, polegające na tym, że do oszacowania wartości możliwych ruchów zamiast zwykłych przeszukiwań losowych używamy zwykłego przeszukiwania Monte Carlo. Dokładniej metoda ta zakłada, że tworząc rozwiązanie przy pomocy NMCS z poziomem  $k$  używamy do wyboru ruchu NMCS z poziomem  $k - 1$  i tak dalej rekurencyjnie aż do poziomu 0, który oznacza przeszukiwanie losowe. Można to przedstawić następującym pseudokodem:

```

NMCS( $v, level$ ){
  while(TRUE){
     $val.push\_back(v)$ ;
    if( $|v.children| == 0$ ) return  $val$ ;
    if( $level == 0$ ){
       $v = random(v.children)$ ;
    }else{
       $best = -1$ ;
      foreach( $u \in v.children$ ){
         $val2 = NMCS(u, level - 1)$ ;
        if( $value(val2) > best$ ){
           $best = value(val2)$ ;
           $move = u$ ;
        }
      }
       $v = u$ ;
    }
  }
}

```

}  
}

### 5.2.2. Uwagi praktyczne

Ze względu na ogromną różnicę w złożoności algorytmu *NMCS* w zależności od użytego poziomu można wprowadzić lekką modyfikację – w przypadku, gdy wyższy poziom jest zbyt kosztowny, zaś niższy za słaby zamiast uruchamiać program z niższym poziomem wiele razy można poszerzyć przeszukiwanie. Poszerzenie takie można łatwo uzyskać uruchamiając dla każdego dziecka symulację poziomu niżej nie raz a  $T$  razy (plus uruchomienie algorytmu najwyższego poziomu tyle razy).

Mając na uwadze algorytm z tą modyfikacją można oszacować pesymistyczną złożoność algorytmu dla szukania długiej izolowanej ścieżki w hiperkostce w zależności od wymiaru  $n$ , poziomu  $L$ , liczby razy przy sprawdzaniu dzieci  $T$  oraz oczekiwanej długości maksymalnej ścieżki  $|P|$ . Złożoność ta jest wyrażona poprzez równanie rekurencyjne:

$$NMCS(n, 0, T, |P|) = n \cdot T \cdot |P|$$

$$NMCS(n, L, T, |P|) = n \cdot T \cdot NMCS(n, L - 1, T, |P| - 1) + NMCS(n, L, T, |P| - 1)$$

Które daje rozwiązanie:

$$NMCS(n, L, T, |P|) = O(T^{L+1} \cdot n^{L+1} \cdot |P|^{L+1})$$

W przypadku szukania izolowanej ścieżki w hiperkostce należy jeszcze doliczyć czas i pamięć potrzebne na sprawdzanie czy nowo wybrany (wylosowany) wierzchołek nie jest sąsiadem jednego z wierzchołków z początku ścieżki.

Jedną z możliwych metod jest za każdym razem przeiterowanie po aktualnej ścieżce i porównanie ciągów binarnych, daje to jednak złożoność czasową rzędu  $n|P|$ . Drugą metodą jest zapamiętywanie wszystkich sąsiadów wierzchołków ze ścieżki (plus ich samych) w tablicy binarnej rozmiaru  $2^n$ . W tym wypadku sprawdzanie jest trywialnie proste (w czasie potrzebnym na odczytanie nazwy wierzchołka), jednak złożoność pamięciowa w przypadku NMCS poziomu 0 przeważa nad złożonością czasową samego algorytmu. W przypadku małych  $n$  nie jest to jednak złe rozwiązanie, ponieważ ze względu na liczne optymalizacje tablic binarnych jest to najszybsze rozwiązanie. Trzecią (pośrednią) metodą jest użycie jako struktury drzewa prefikсового lub tablicy hashującej uzyskując podobnie do tablicy binarnej złożoność  $O(n)$  na odczytanie i wstawienie wierzchołka jednocześnie potrzebując jedynie  $O(n^{1.5} \cdot |P|)$  lub  $O(n \cdot |P|)$  pamięci w zależności od użytej struktury.

W przypadku użycia struktury zapamiętującej sąsiadów wierzchołków ze ścieżki należy jeszcze zwrócić uwagę na zmienianie jej stanu przy wchodzeniu i wracaniu z niższych poziomów wywołania. Można albo za każdym razem pamiętać wszystkie wstawione wierzchołki i przy zwracaniu wyniku do poziomu wyżej dokonywać ich usunięcia, albo można za każdym razem kopiować całą strukturę (dosyć efektywne przy użyciu tablic). W drugim przypadku wystarczy mieć  $L + 1$  kopii struktury na raz (po jednej na każdy poziom) i przy powrocie z poziomu niżej zastępować starą wersję z tamtego poziomu kopią z poziomu o 1 wyżej.

Mając to na uwadze można zaprogramować algorytm tak, aby miał złożoności czasową  $O(T^{L+1} \cdot n^{L+2} \cdot |P|^{L+1})$  i pamięciową  $O(L \cdot n \cdot |P|)$ .

Przeszukiwanie można dodatkowo zawęzić jeśli program weźmie pod uwagę, że kierunki, które nie pojawiły się jeszcze na ścieżce są równoważne (wystarczy pamiętać liczbę oznaczającą ilość kierunków już użytych i nie używać kierunków większych niż ta liczba + 1). W ten sposób ustalone są pierwsze 3 ruchy ( $1 \rightarrow 2 \rightarrow 4 \rightarrow 8$  przy numerowaniu klasycznym), zaś na czwarty są tylko dwie możliwości ( $8 \rightarrow 7, 8 \rightarrow 17$ ). Daje to duże przyśpieszenie przy małych przypadkach, jednak przy większych nie ma już większego znaczenia.

### 5.2.3. Uzyskane wartości

Przy pomocy komputera i przedstawionych wyżej algorytmów przeprowadziłem przeszukiwanie w celu znalezienia możliwie najdłuższej takiej ścieżki. W przypadku  $n \leq 6$  przy pomocy pełnego przeszukiwania, a dalej NMCS z 4 poziomami dla  $n = 7, 3$  dla  $n = 8, 2$  dla  $9 \leq n \leq 12$  i 1 dla  $13 \leq n \leq 18$  z ilością wywołań niższego poziomu takich żeby podobny był czas dla wszystkich  $n$ . Dalej użyłem zwykłego błędzenia losowego i wybierania najdłuższej uzyskanej tak ścieżki (ilość przeszukiwań znów dostosowana do tego czasu). Program użyty do takiego przeszukiwania wraz z zapisami ścieżek dla mniejszych  $n$  (dla większych brak ze względu na wielkość plików) i dokładnymi parametrami jest dołączony do pracy.

Poniższa tabelka i wykres logarymiczny prezentują porównanie najlepszych rezultatów uzyskanych dla poszczególnych  $n$  ( $|P|$ ) do wielkości ścieżki Fibonacciego i całego grafu.

| $n$                 | 0 | 1 | 2    | 3    | 4    | 5   | 6    | 7    | 8    | 9    | 10   | 11   | 12   | 13   |
|---------------------|---|---|------|------|------|-----|------|------|------|------|------|------|------|------|
| $F_{n+1}$           | 1 | 2 | 3    | 5    | 8    | 13  | 21   | 34   | 55   | 89   | 144  | 233  | 377  | 610  |
| $ P  + 1$           | 1 | 2 | 3    | 5    | 8    | 14  | 27   | 51   | 86   | 146  | 245  | 423  | 749  | 1373 |
| $2^n$               | 1 | 2 | 4    | 8    | 16   | 32  | 64   | 128  | 256  | 512  | 1024 | 2048 | 4096 | 8192 |
| $\sqrt[n]{ P  + 1}$ |   | 2 | 1.73 | 1.71 | 1.68 | 1.7 | 1.73 | 1.75 | 1.75 | 1.74 | 1.73 | 1.73 | 1.74 | 1.74 |

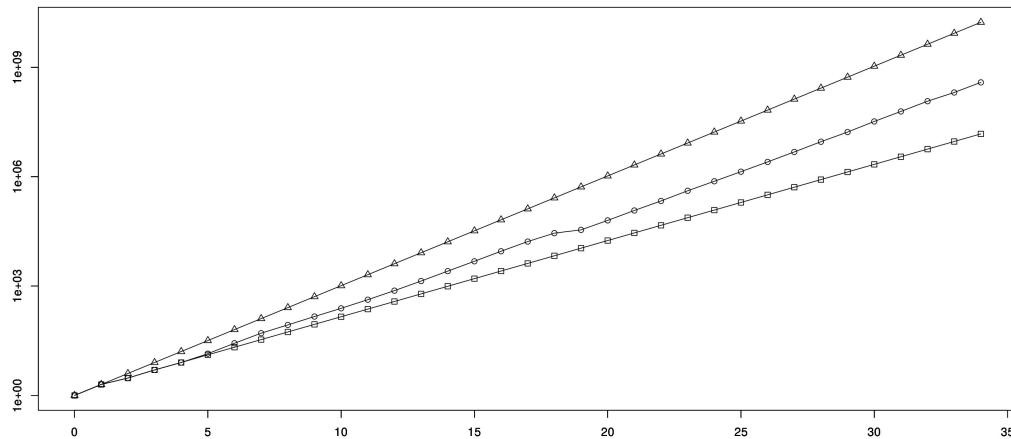
| $n$                 | 14    | 15    | 16    | 17     | 18     | 19     | 20      | 21      | 22      |
|---------------------|-------|-------|-------|--------|--------|--------|---------|---------|---------|
| $F_{n+1}$           | 987   | 1597  | 2584  | 4181   | 6765   | 10946  | 17711   | 28657   | 46368   |
| $ P  + 1$           | 2568  | 4778  | 9017  | 16612  | 28287  | 34801  | 63271   | 118344  | 216033  |
| $2^n$               | 16384 | 32768 | 65536 | 131072 | 262144 | 524288 | 1048576 | 2097152 | 4194304 |
| $\sqrt[n]{ P  + 1}$ | 1.75  | 1.76  | 1.77  | 1.77   | 1.77   | 1.73   | 1.74    | 1.74    | 1.75    |

| $n$                 | 23      | 24       | 25       | 26       | 27        | 28        | 29        |
|---------------------|---------|----------|----------|----------|-----------|-----------|-----------|
| $F_{n+1}$           | 75025   | 121393   | 196418   | 317811   | 514229    | 832040    | 1346269   |
| $ P  + 1$           | 410719  | 748502   | 1373384  | 2543441  | 4774783   | 9098482   | 16788856  |
| $2^n$               | 8388608 | 16777216 | 33554432 | 67108864 | 134217728 | 268435456 | 536870912 |
| $\sqrt[n]{ P  + 1}$ | 1.75    | 1.76     | 1.76     | 1.76     | 1.77      | 1.77      | 1.77      |

| $n$                 | 30         | 31         | 32         | 33         | 34          |
|---------------------|------------|------------|------------|------------|-------------|
| $F_{n+1}$           | 2178309    | 3524578    | 5702887    | 9227465    | 14930352    |
| $ P  + 1$           | 32747927   | 61637291   | 117676035  | 204031449  | 386051791   |
| $2^n$               | 1073741824 | 2147483648 | 4294967296 | 8589934592 | 17179869184 |
| $\sqrt[n]{ P  + 1}$ | 1.78       | 1.78       | 1.79       | 1.79       | 1.79        |







# Bibliografia

- [1] L. Lovasz, J. Pelikan and K. Vesztergombi. "Discrete Mathematics, Elementary and Beyond." *Undergraduate Texts in Mathematics*. New York: Springer, first edition, 2003
- [2] L. H. HARPER, "Optimal Numberings and Isoperimetric Problems on Graphs" *JOURNAL OF COMBINATORIAL THEORY* 1, 385-393 (1966)
- [3] Tomas Dvorak, Jiri Fink, Petr Gregor, Vaclav Koubek and Tomasz Radzik, "Efficient connectivity testing of hypercubic networks with faults"
- [4] Jiri Fink and Petr Gregor, "Long paths and cycles in hypercubes with faulty vertices"
- [5] G. Wiener, "Edge multiplicity and other trace functions." *In Proceedings of European Conference on Combinatorics, Graph Theory and Applications (EuroComb 2007), volume 29 of Electronic Notes in Discrete Mathematics, pages 491-495, 2007.*
- [6] Jiri Fink and Petr Gregor, "Long pairs of paths in faulty hypercubes"
- [7] Nelson Casenada and Ivan S. Gotchev, "Proof of Locke's Conjecture, I"
- [8] David Pegrimek, "Hamiltonian cycles in hypercubes with removed vertices", *Prague 2013*
- [9] Sun, Chao-Ming and Jou, Yue-Dar, "Hamiltonian laceability of hypercubes with some faulty elements", *In Proceedings of 2009 IEEE International Conference on Networking, Sensing and Control, Okayama, Japan, March 26-29, 2009, pp.626-630*
- [10] Tristan Cazenave, "Nested Monte-Carlo Search" *LAMSADE Universite Paris-Dauphine Paris, France*
- [11] Frank Harary, John P. Hayes and Horng-Jyh Wu, "A survey of theory of hypercube graphs" *Comput. Math. Applic. Vol. 15, No 4, pp. 277-289, 1988*
- [12] Frank Harary, Marilynn Livingston, "Independent domination in hypercubes" *Appl. Math. Lett. Vol. 6, No 3, pp. 27-28, 1993*
- [13] Wojciech Rytter & Bartosz Szreder, "Wprowadzenie do kombinatoryki algorytmicznej"
- [14] DONALD E. KNUTH, "Generating All Tuples and Permutations" *THE ART OF COMPUTER PROGRAMMING VOLUME 4, FASCICLE 2*
- [15] FRANK RUSKEY, "Combinatorial Generation" *Working Version, October 1, 2003*
- [16] Tibor Szabo, Emo Wertz, "Unique Sink Orientations of Cubes"
- [17] Chi Him Wong, "Novel universal cycle constructions for a variety of combinatorial objects" *Guelph, Ontario, Canada, April, 2015*