Algorytmy Macierzowe

Sprawozdanie z laboratorium nr.3

Władysław Jerzy Nieć, Paweł Surdyka

Zadania

Proszę napisać algorytmy permutacji macierzy

- Minimum degree (10 punktów) [dla dużej macierzy można zmodyfikować do AMD]
- Cuthill-McKee (10 punktów)
- Reversed Cuthill-McKee [odwrócona permutacja Cuthill-McKee (fajny algorytm)] (5 punktów)

Algorytm Minimum degree

```python
def minimum_degree(matrix):
    # Pobranie wymiarów macierzy
    m, n = matrix.shape

    # Inicjalizacja pustej permutacji
    permutation = []

    # Utworzenie słownika sąsiedztwa na podstawie macierzy
    adjacency = neighborhood_dict(matrix)

    # Iteracja po wszystkich wierzchołkach grafu
    for i in range(n):
        # Znalezienie wierzchołka o najmniejszym stopniu
        min_degree, best_vertex = min(((len(adjacent), v) for v, adjacent in adjacency.items()), key=lambda x: x[0])

        # Usunięcie wierzchołka o najmniejszym stopniu z listy sąsiedztwa pozostałych wierzchołków
        for vertex in adjacency:
            adjacency[vertex].discard(best_vertex)

        # Aktualizacja listy sąsiedztwa sąsiadów wierzchołka o najmniejszym stopniu
        for neighbor in adjacency[best_vertex]:
            adjacency[neighbor].update(adjacency[best_vertex].difference(set([neighbor])))

        # Usunięcie wierzchołka o najmniejszym stopniu z listy sąsiedztwa
        del adjacency[best_vertex]

        # Dodanie wierzchołka o najmniejszym stopniu do permutacji
        permutation.append(best_vertex)

    # Zastosowanie permutacji do macierzy sąsiedztwa i zwrócenie wyniku
    return apply_permutation(matrix, permutation)
```

Znajdowanie sąsiadów

```python
def neighborhood_dict(matrix):#Zwraca słownik par wierzchołek -> zbiór jego sąsiadów
    m, n = matrix.shape
    return {i: set(j for j in range(n) if matrix[i][j] > 0 and i != j) for i in range(m)}
```

Permutowanie

```python
def apply_permutation(matrix, permutation):
    matrix = np.array([matrix[i, :] for i in permutation])
    return np.array([matrix[:, i] for i in permutation])
```

Algorytm Cuthill-McKee (zwykły i odwrócony)

BFS

```python
def cuthill_mckee_bfs(adjacency, visited, vertex, permutation):
    # Kolejka do przechowywania wierzchołków
    q = SimpleQueue()

    # Rozpoczęcie BFS od danego wierzchołka
    q.put(vertex)

    while not q.empty():
        vertex = q.get()

        # Jeśli wierzchołek nie został odwiedzony
        if not visited[vertex]:
            visited[vertex] = True

            # Dodanie wierzchołka do permutacji
            permutation.append(vertex)

            # Sąsiedzi posortowani rosnąco według stopnia
            for neighbor in sorted(list(adjacency[vertex]), key=lambda x: len(adjacency[x])):
                # Jeśli sąsiad nie został jeszcze odwiedzony, dodaj go do kolejki
                if not visited[neighbor]:
                    q.put(neighbor)
```

Znalezienie permutacji

```python
def cuthill_mckee_permutation(matrix):
    m, n = matrix.shape

    # Utworzenie słownika sąsiedztwa na podstawie macierzy
    adjacency = neighborhood_dict(matrix)

    # Posortowanie wierzchołków według stopnia
    sorted_vertices = sorted([(vertex, len(neighbors)) for vertex, neighbors in adjacency.items()], key=lambda x: x[1])

    # Inicjalizacja pustej permutacji
    permutation = []

    # Lista odwiedzonych wierzchołków
    visited = [False for i in range(m)]

    # Iteracja po posortowanych wierzchołkach
    for vertex, _ in sorted_vertices:
        # Jeśli wierzchołek nie został odwiedzony, wykonaj BFS
        if not visited[vertex]:
            cuthill_mckee_bfs(adjacency, visited, vertex, permutation)

    # Zwrócenie permutacji
    return permutation
```

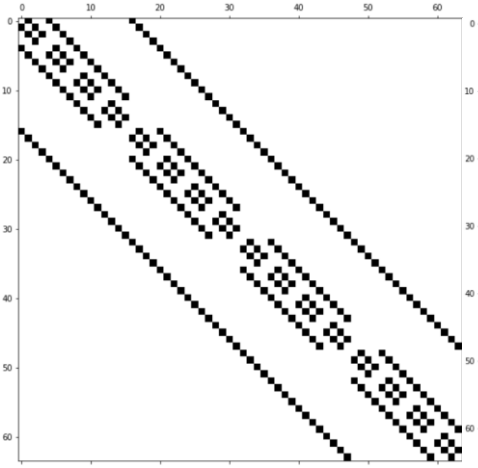
```python
def cuthill_mckee(matrix):
    # Zastosowanie permutacji do macierzy sąsiedztwa i zwrócenie wyniku
    return apply_permutation(matrix, cuthill_mckee_permutation(matrix))
```

W wersji odwróconej

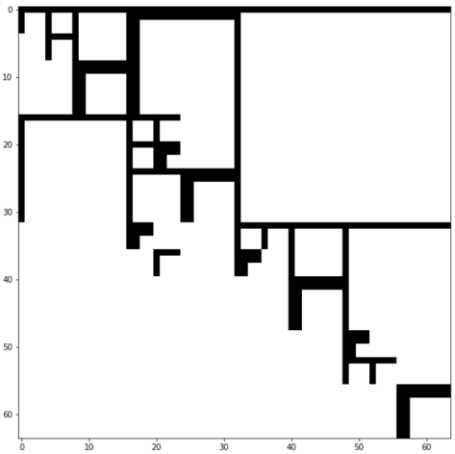
```python
def reversed_cuthill_mckee(matrix):
    return apply_permutation(matrix, cuthill_mckee_permutation(matrix)[::-1])
```

Pomiary i wizualizacja

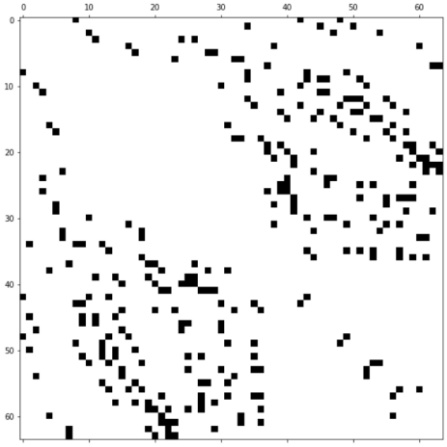
Dla k = 2 (tj. rozmiaru macierzy 2^6), δ = size/2, b = 2



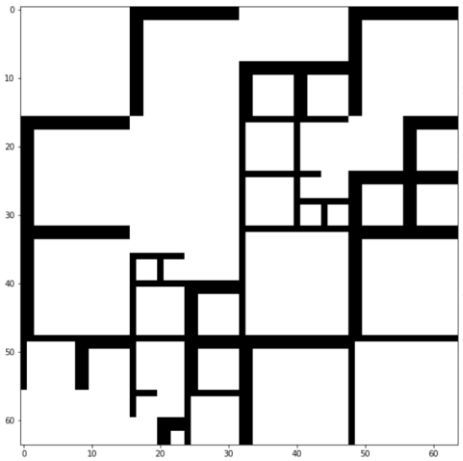
przed kompresją



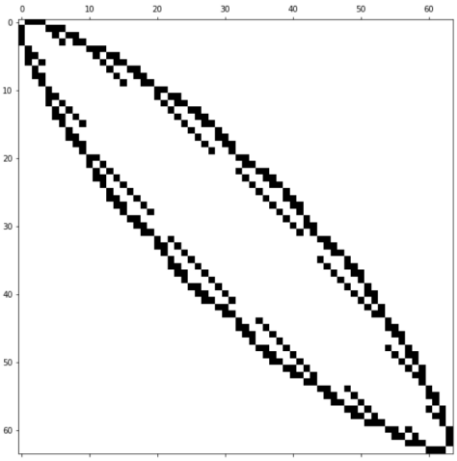
po kompresji bez permutacji



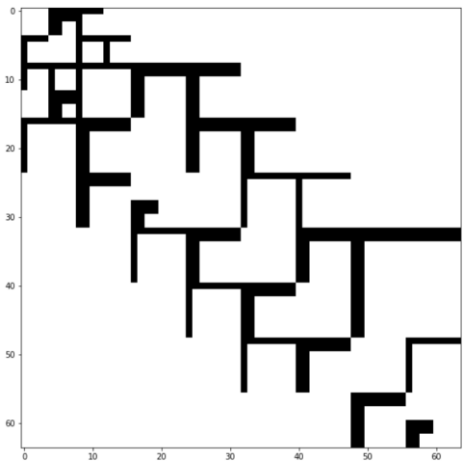
po permutacji minimum_degree



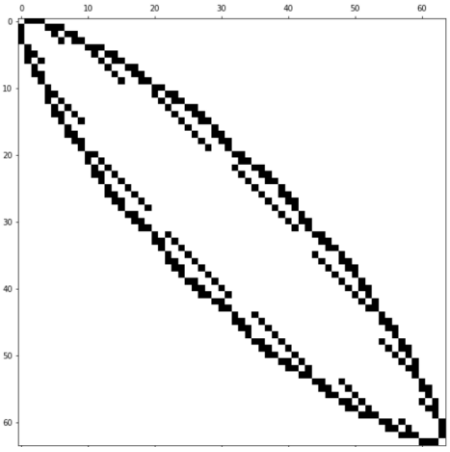
po kompresji (minimum_degree)



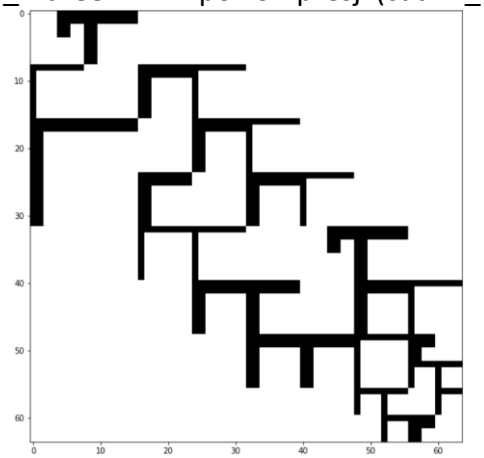
po permutacji cuthill_mckee



po kompresji (cuthill_mckee)



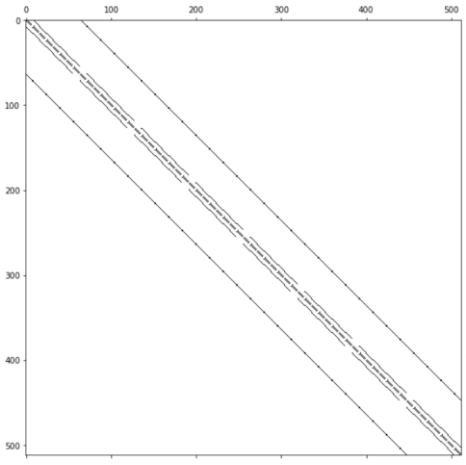
po permutacji reversed_cuthill_mckee



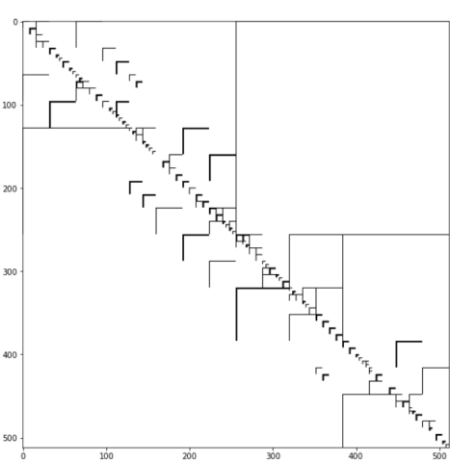
po kompresji (reversed_cuthill_mckee)

```
Rozmiar oryginalny:     32768B
Rozmiar macierzy rzadkiej:      3716B
Rozmiar macierzy hierarchicznej bez permutacji: 17760B
Rozmiar macierzy hierarchicznej z permutacją minimum_degree:    19872B
Rozmiar macierzy hierarchicznej z permutacją cuthill_mckee:     22936B
Rozmiar macierzy hierarchicznej z permutacją reversed_cuthill_mckee:    22936B
```

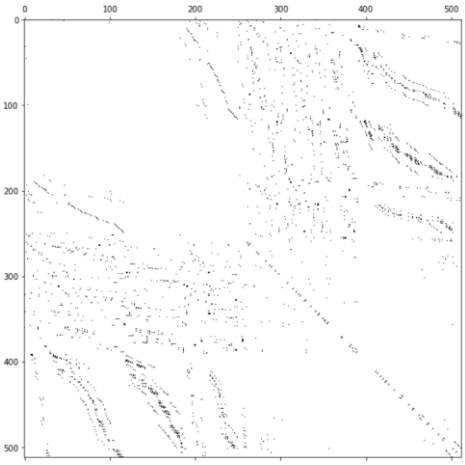
Dla k = 3 (tj. rozmiaru macierzy 2^9)



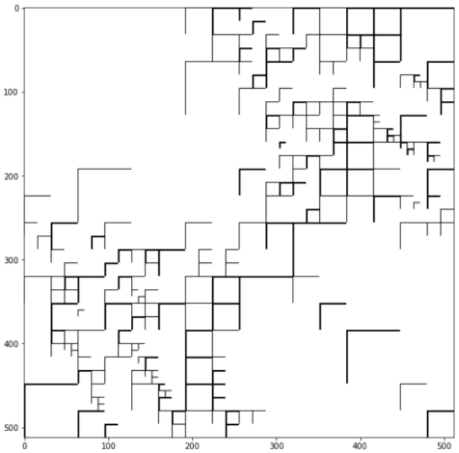
przed kompresją



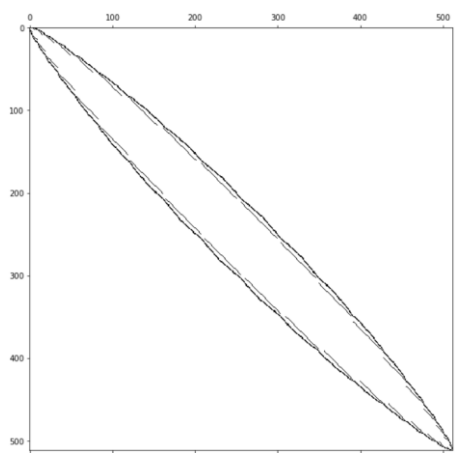
po kompresji bez permutacji



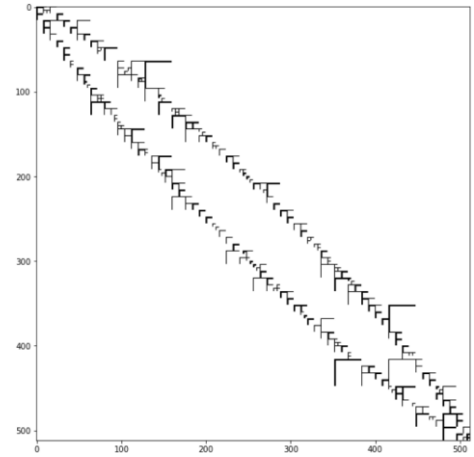
po permutacji minimum_degre



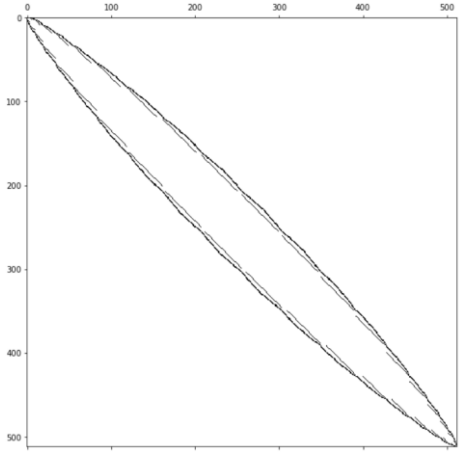
po kompresji (minimum_degree)



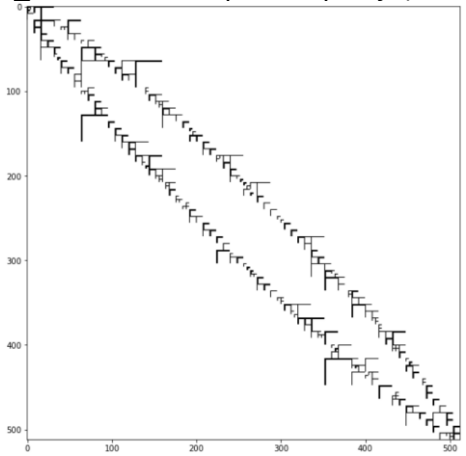
po permutacji cuthill_mckee



po kompresji (cuthill_mckee)



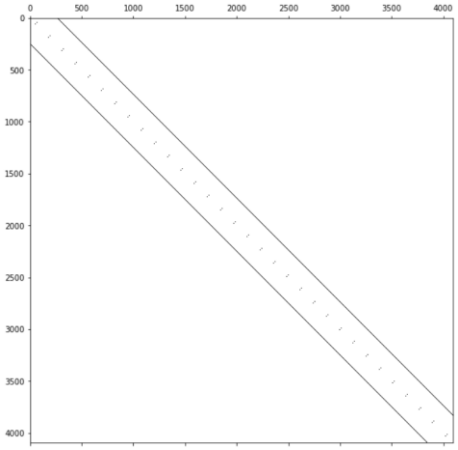
po permutacji reversed_cuthill_mckee



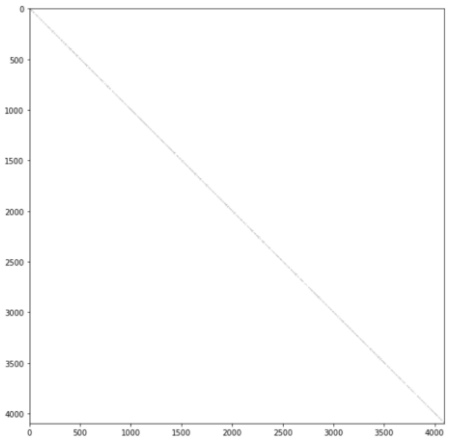
po kompresji (reversed_cuthill_mckee)

```
Rozmiar oryginalny:     2097152B
Rozmiar macierzy rzadkiej:      34308B
Rozmiar macierzy hierarchicznej bez permutacji: 125256B
Rozmiar macierzy hierarchicznej z permutacją minimum_degree:    182920B
Rozmiar macierzy hierarchicznej z permutacją cuthill_mckee:     219872B
Rozmiar macierzy hierarchicznej z permutacją reversed_cuthill_mckee:    219872B
```

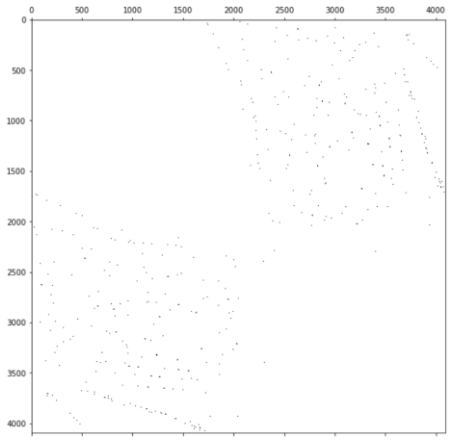
Dla k = 4 (tj. rozmiaru macierzy 2^12)



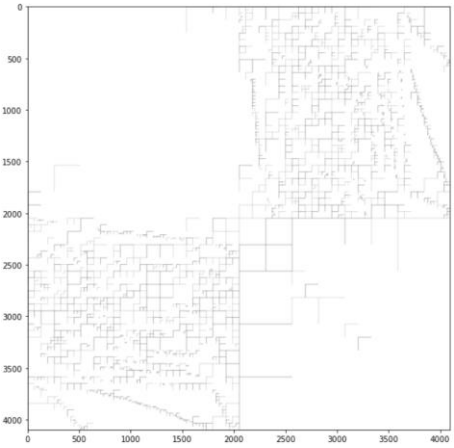
przed kompresją



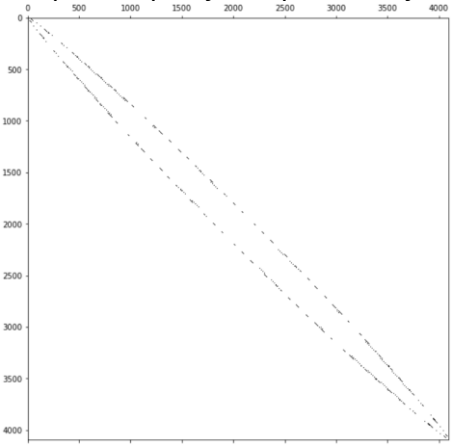
po kompresji bez permutacji



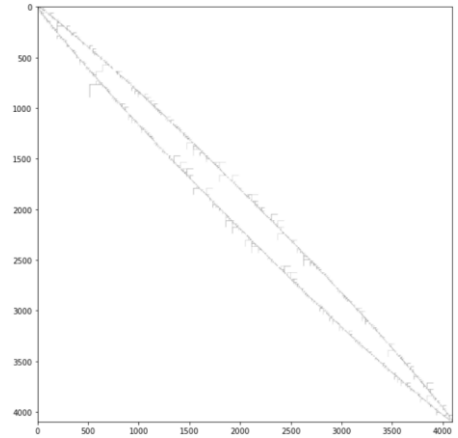
po permutacji minimum_degree



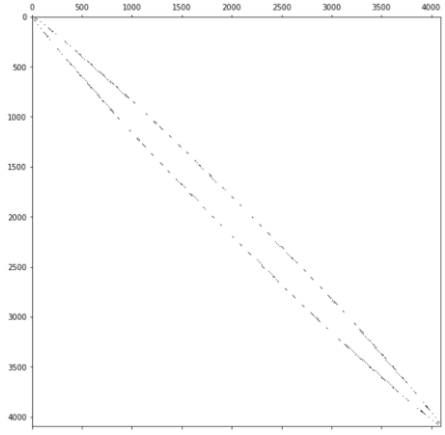
po kompresji (minimum_degree)



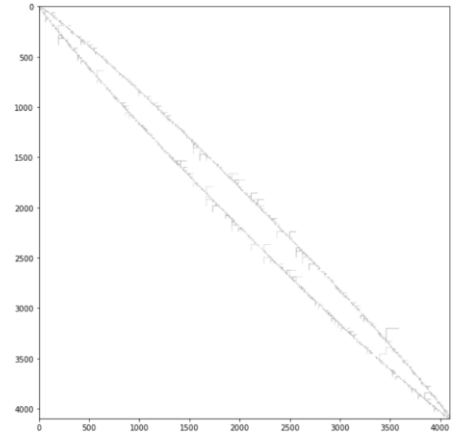
po permutacji cuthill_mckee



po kompresji (cuthill_mckee)



po permutacji reversed_cuthill_mckee

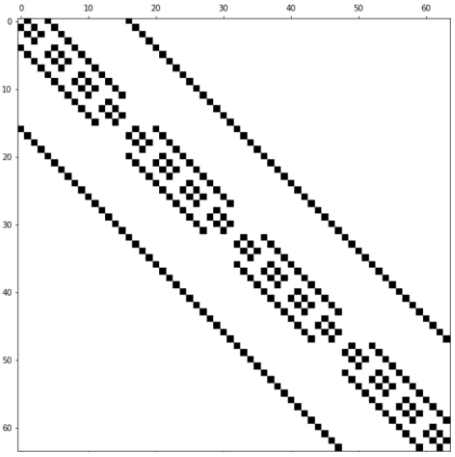


po kompresji (reversed_cuthill_mckee)

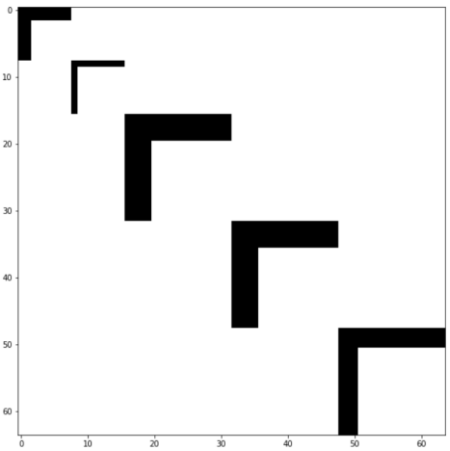
```
Rozmiar oryginalny:     134217728B
Rozmiar macierzy rzadkiej:      292868B
Rozmiar macierzy hierarchicznej bez permutacji: 612616B

Rozmiar macierzy hierarchicznej z permutacją minimum_degree:    1942216B
Rozmiar macierzy hierarchicznej z permutacją cuthill_mckee:     1521296B
Rozmiar macierzy hierarchicznej z permutacją reversed_cuthill_mckee:    1521296B
```

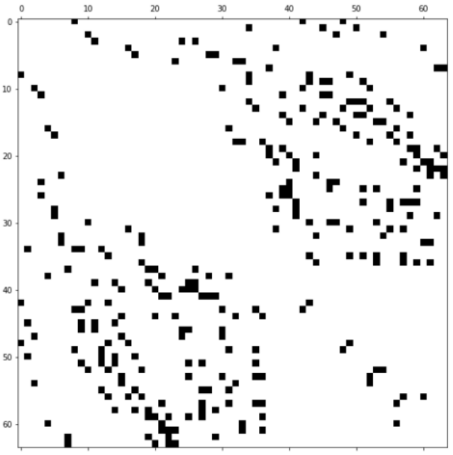
k = 2 (tj. rozmiaru macierzy 2^6), δ = size/3, b = 4



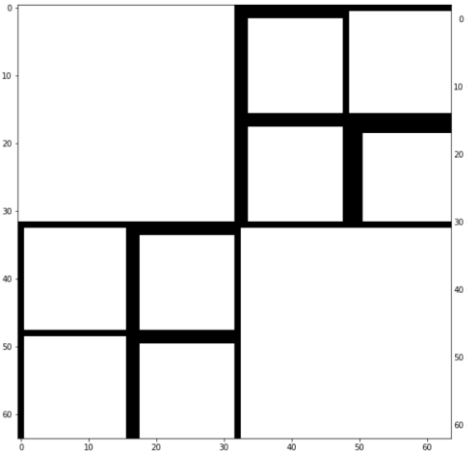
przed kompresją



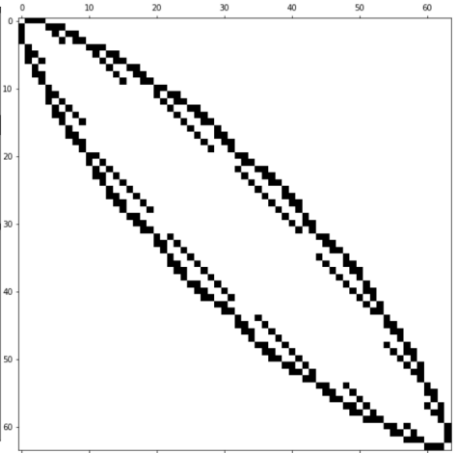
po kompresji bez permutacji



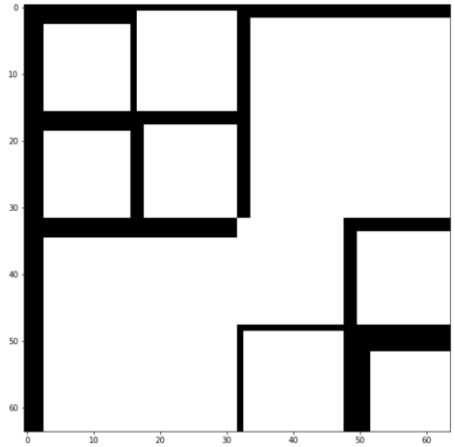
po permutacji minimum_degree



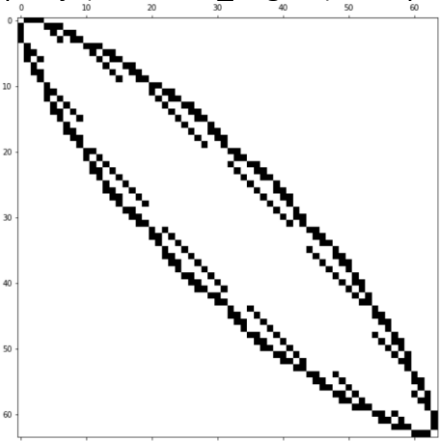
po kompresji (minimum_degree)



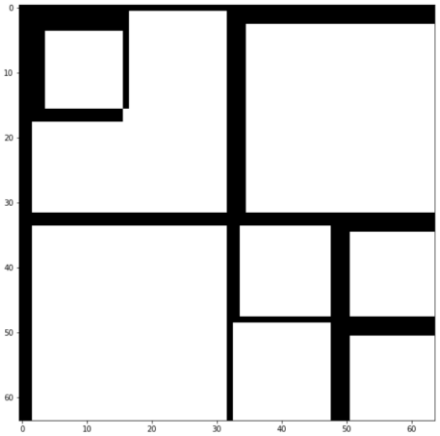
po permutacji cuthill_mckee



po kompresji (cuthill_mckee)



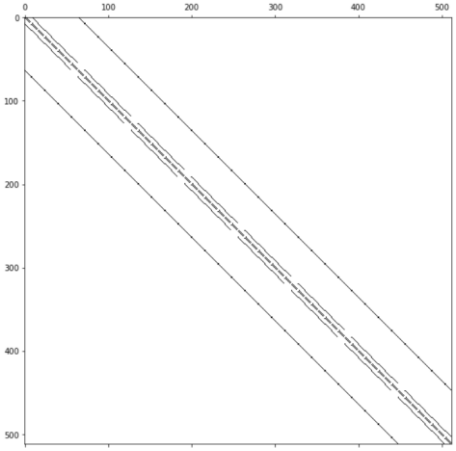
po permutacji reversed_cuthill_mckee



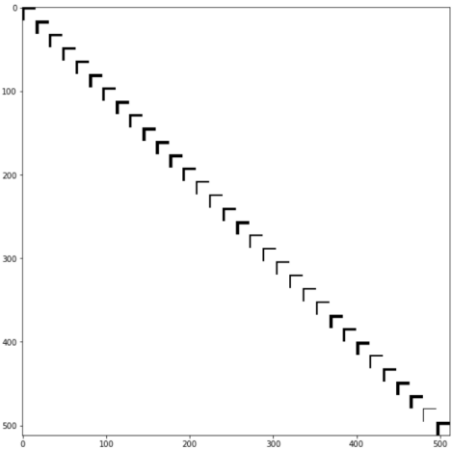
po kompresji (reversed_cuthill_mckee)

```
Rozmiar oryginalny:     32768B
Rozmiar macierzy rzadkiej:      3716B
Rozmiar macierzy hierarchicznej bez permutacji: 5832B
Rozmiar macierzy hierarchicznej z permutacją minimum_degree:    6160B
Rozmiar macierzy hierarchicznej z permutacją cuthill_mckee:     7440B
Rozmiar macierzy hierarchicznej z permutacją reversed_cuthill_mckee:    7440B
```

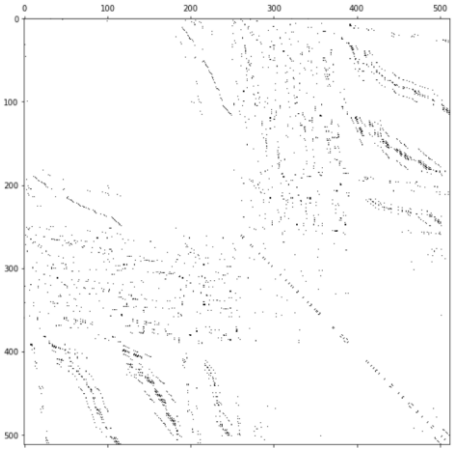
k = 3 (tj. rozmiaru macierzy 2^9), δ = size/3, b = 4



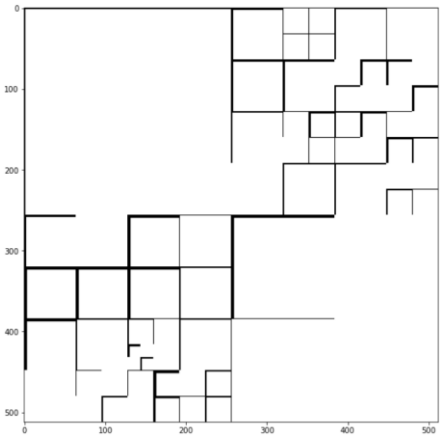
przed kompresją



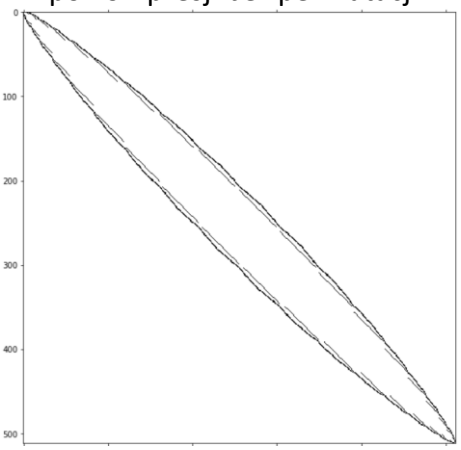
po kompresji bez permutacji



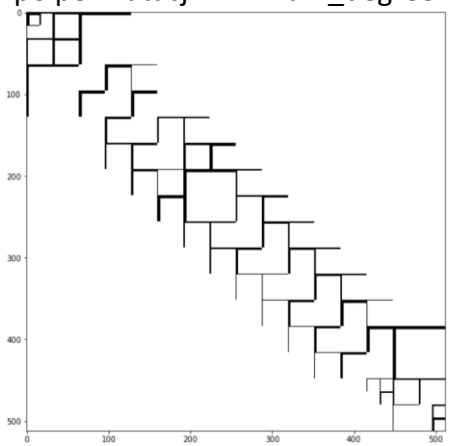
po permutacji minimum_degree



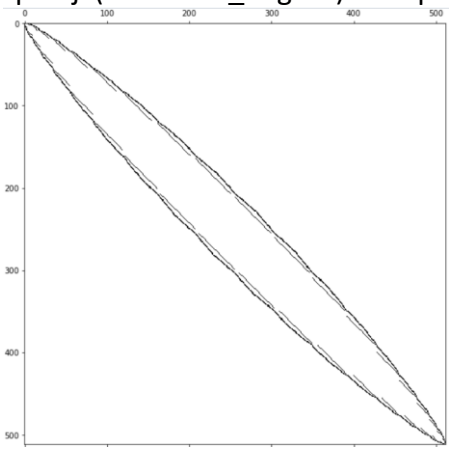
po kompresji (minimum_degree)



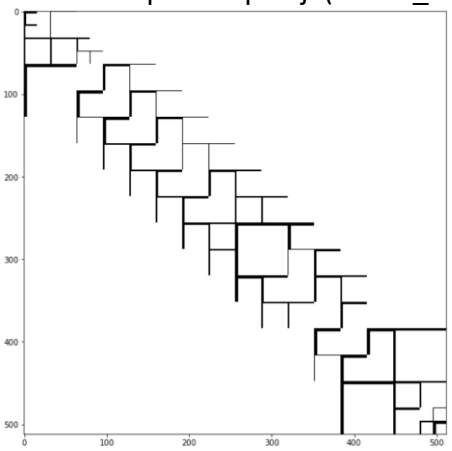
po permutacji cuthill_mckee



po kompresji (cuthill_mckee)



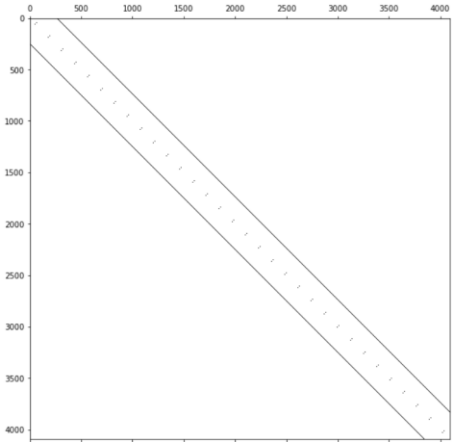
po permutacji reversed_cuthill_mckee



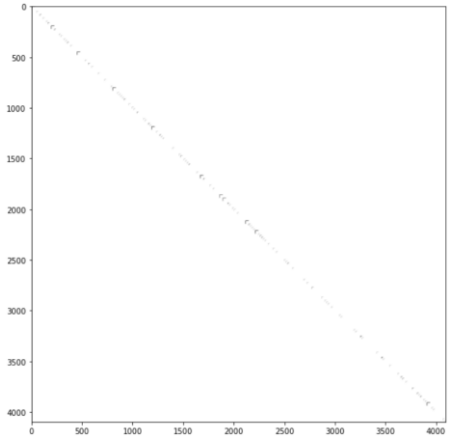
po kompresji (reversed_cuthill_mckee)

```
Rozmiar oryginalny:     2097152B
Rozmiar macierzy rzadkiej:      34308B
Rozmiar macierzy hierarchicznej bez permutacji: 42568B
Rozmiar macierzy hierarchicznej z permutacją minimum_degree:    76224B
Rozmiar macierzy hierarchicznej z permutacją cuthill_mckee:     75264B
Rozmiar macierzy hierarchicznej z permutacją reversed_cuthill_mckee:    75264B
```

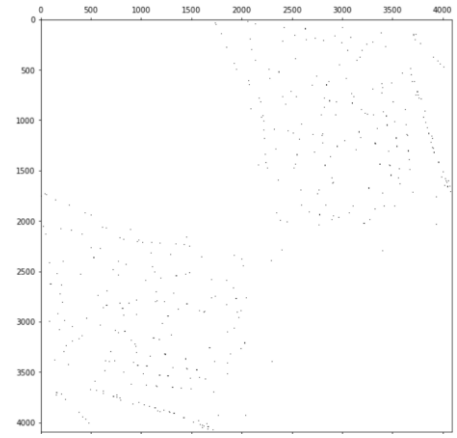
k = 4 (tj. rozmiaru macierzy 2^12), δ = size/3, b = 4



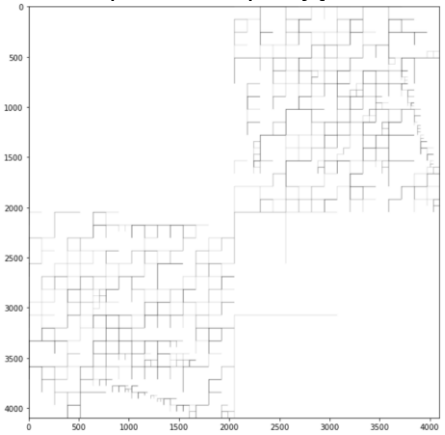
przed kompresją



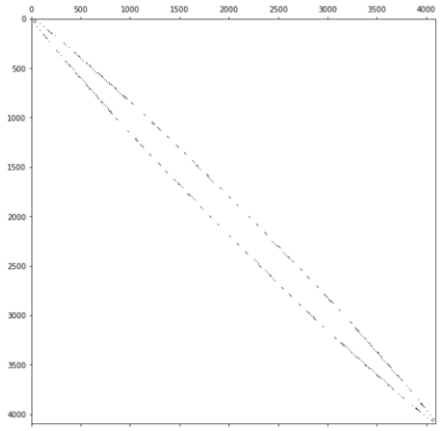
po kompresji bez permutacji



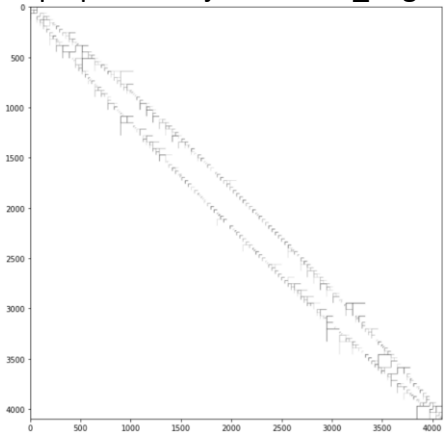
po permutacji minimum_degree



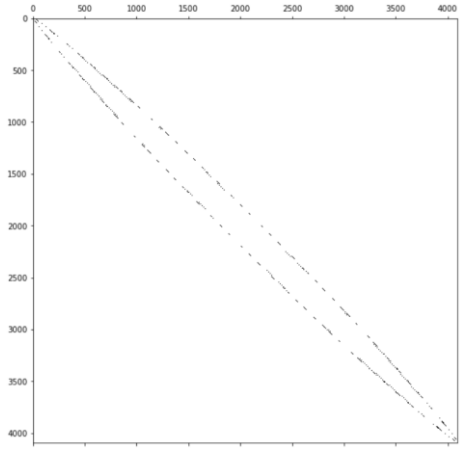
po kompresji (minimum_degree)



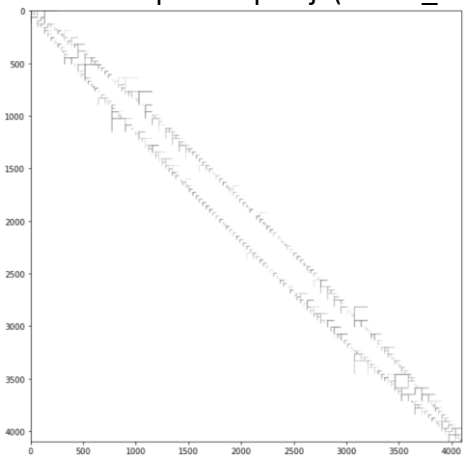
po permutacji cuthill_mckee



po kompresji (cuthill_mckee)



po permutacji reversed_cuthill_mckee



po kompresji (reversed_cuthill_mckee)

```
Rozmiar oryginalny:     134217728B
Rozmiar macierzy rzadkiej:      292868B
Rozmiar macierzy hierarchicznej bez permutacji: 238576B
Rozmiar macierzy hierarchicznej z permutacją minimum_degree:    1113144B
Rozmiar macierzy hierarchicznej z permutacją cuthill_mckee:     622264B
Rozmiar macierzy hierarchicznej z permutacją reversed_cuthill_mckee:    622264B
```

δ = size/2 , b = 2

Rozmiar\parametry	k = 2	k = 3	k = 4
orginalny	32768B	2097152B	134217728B
macierzy rzadkiej	3716B	34308B	292868B
macierzy hierarchicznej bez permutacji	17760B	125256B	612616B
macierzy hierarchicznej z permutacją minimum_degree	19872B	182920B	1942216B
macierzy hierarchicznej z permutacją cuthill_mckee	22936B	219872B	1521296B
macierzy hierarchicznej z permutacją reversed_cuthill_mckee	22936B	219872B	1521296B

δ = size/3 , b = 4

Rozmiar\parametry	k = 2	k = 3	k = 4
orginalny	32768B	2097152B	134217728B
macierzy rzadkiej	3716B	34308B	292868B
macierzy hierarchicznej bez permutacji	5832B	42568B	238576B
macierzy hierarchicznej z permutacją minimum_degree	6160B	76224B	1113144B
macierzy hierarchicznej z permutacją cuthill_mckee	7440B	75264B	622264B
macierzy hierarchicznej z permutacją reversed_cuthill_mckee	7440B	75264B	622264B

Wnioski

- Niema różnicy w jakości kompresji pomiędzy permutacją cuthill_mckee a reversed_cuthill_mckee.

- Im mniej wartości osobliwych tym lepsza kompresja.

- Dla mniejszych macierzy permutacja minimum_degree daje lepsze efekty a dla większych cuthill_mckee.