Politechnika Krakowska
Wydział Inżynierii Elektrycznej i Komputerowej
Katedra Informatyki Technicznej

Instrukcja do ćwiczeń laboratoryjnych
z przedmiotu:

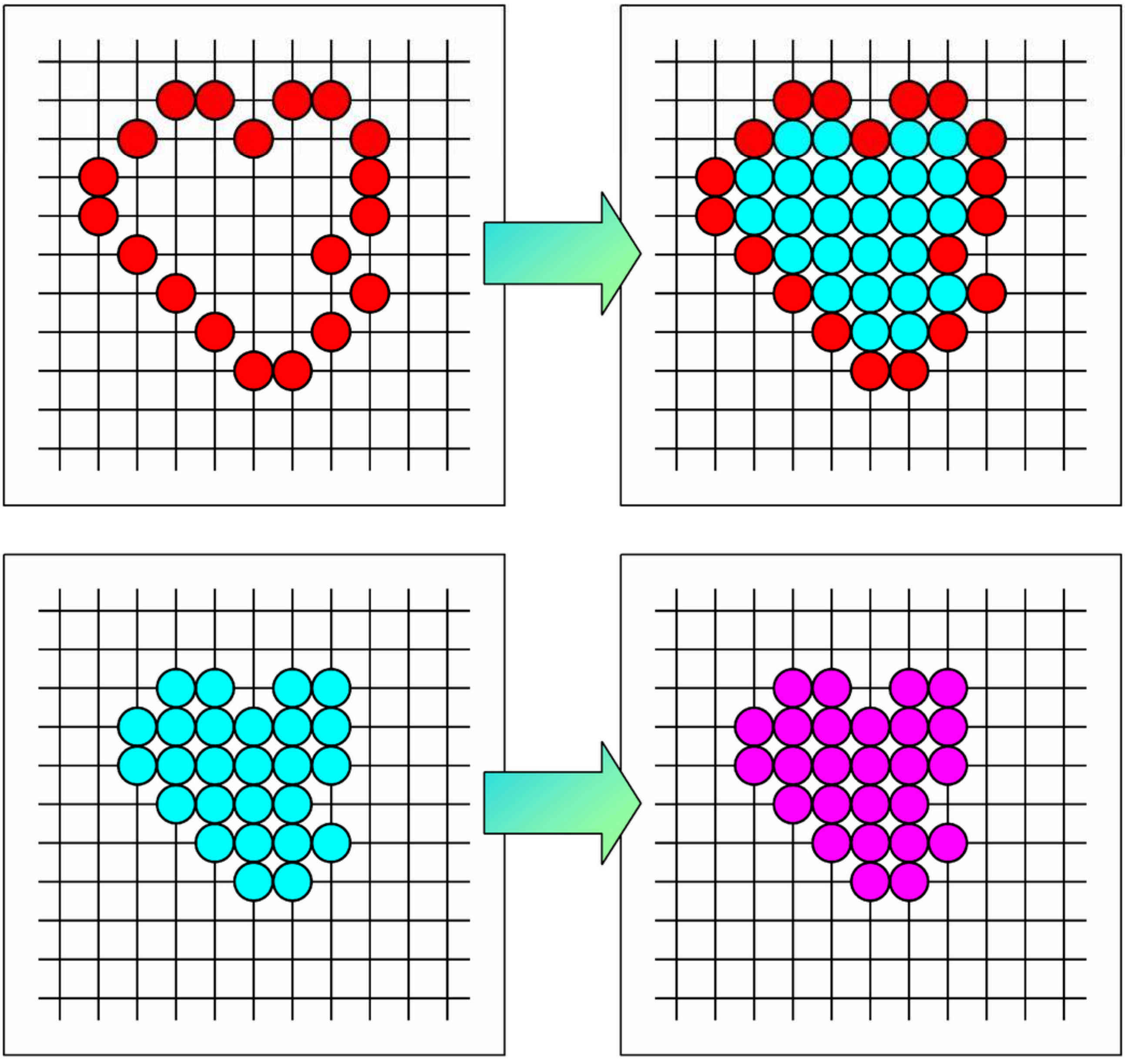
Grafika Komputerowa i Multimedia

Wypełnianie obszaru

Podstawy

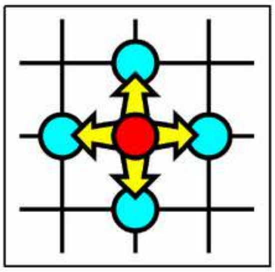
Wypełnianie obszaru jest drugim po rasteryzacji odcinka lub okręgu, najczęściej występującym problemem związanym z prymitywami w grafice rastrowej. Rozwiązanie dla szczególnych przypadków (np. dla prostokąta) jest kwestią wręcz trywialną. W ogólnym przypadku algorytm powinien działać poprawnie dla dowolnego wielokąta (także wklęsłego), jak i dla wielokątów z „dziurami".

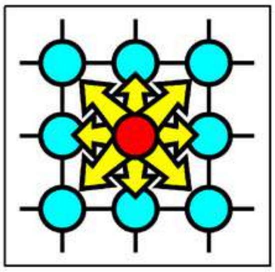
Istnieje analogia pomiędzy wypełnieniem obszaru, a zmianą barwy w danym obszarze. Algorytm wypełniania obszaru może posłużyć do zmiany barwy w danym obszarze.



Spójność

Zbiór pikseli jest spójny, jeżeli z dowolnego piksela z tego zbioru można przejść do każdego innego przez piksele sąsiednie.

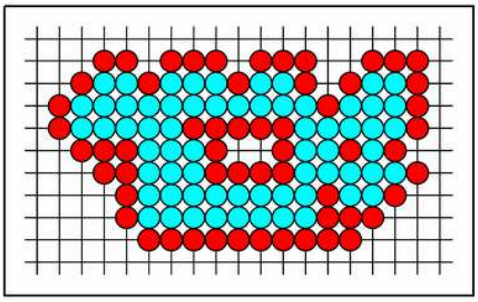




- gdy sąsiadów 4: zbiór czterospójny
- gdy sąsiadów 8: zbiór ośmiospójny

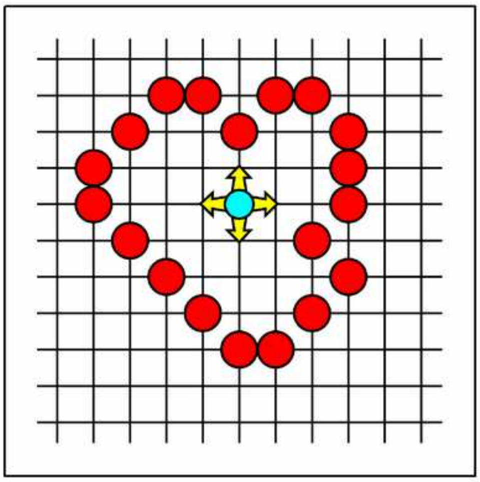
Wypełnianie przez spójność

Najczęściej rozważany przypadek, to obszar będący zbiorem czterospójnym, a ograniczający brzeg – ośmiospójnym. Dopuszczamy możliwość istnienia „dziur" wewnątrz obszaru – będą to podobszary ograniczone ośmiospójnymi brzegami pikseli. Szczególny przypadek „dziur" to pojedyncze piksele w kolorze brzegu.



Znając ziarno, czyli piksel leżący wewnątrz obszaru, możemy wypełnić ten obszar nowym kolorem i próbować „siać" ten kolor (propagować przez spójność) w czterech kierunkach. Siać, tzn. sprawdzać, czy cztery sąsiednie piksele należą do wnętrza obszaru i czy nie zostały jeszcze wypełnione nowym kolorem. Dalej postępujemy analogicznie, badając piksele sąsiadujące z sąsiadami ziarna.

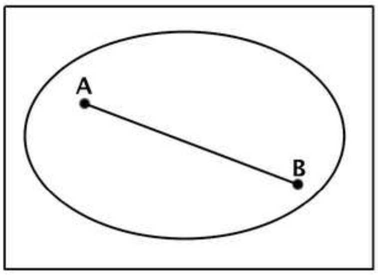
```
procedure spójność4(x,y)
begin
    putpixel(x,y,kolorWypełnienia);
    if (kolor(x-1,y) inny niż kolorBrzegu i inny
    niż kolorWypełnienia) spójność4(x-1,y);
    if (kolor(x+1,y) inny niż kolorBrzegu i inny
    niż kolorWypełnienia) spójność4(x+1,y);
    if (kolor(x,y-1) inny niż kolorBrzegu i inny
    niż kolorWypełnienia) spójność4(x,y-1);
    if (kolor(x,y+1) inny niż kolorBrzegu i inny
    niż kolorWypełnienia) spójność4(x,y+1);
end
```



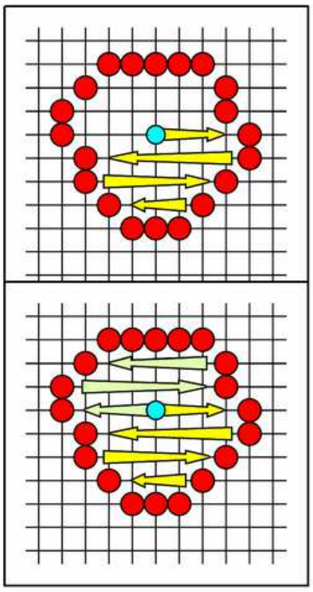
Algorytm jest prosty, jednakże jego praktyczna realizacja może być trudna i kosztowna. Przy dużym obszarze głębokość rekurencji jest zazwyczaj znaczna, a wtedy częstokroć następuje przepełnienie budowanego stosu. Algorytm jest rozrzutny – kolor tego samego piksela bada się kilka razy.

Wypełnianie obszarów wypukłych

W przypadku, gdy obszar jest wypukły, nie zawiera „dziur” możemy zastosować inny algorytm. Obszar jest wypukły, gdy dla dowolnej pary punktów A, B leżących wewnątrz brzegu odcinek, którego końcami są te punkty leży w całości wewnątrz brzegu.

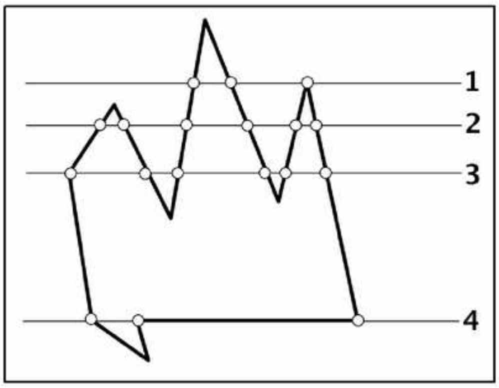


- Wypełniamy w linii rozpoczynając od punktu startowego, aż do prawej granicy brzegu
- Znajdujemy „niższy” punkt brzegu i wypełniamy linię do lewej granicy brzegu
- Powtarzamy do momentu, aż nie możemy znaleźć punktu „niższego”
- Wracamy do punktu startowego i kontynuujemy proces idąc w kierunku lewej granicy brzegu
- Znajdujemy „wyższy” punkt brzegu i wypełniamy linię do prawej granicy brzegu
- Powtarzamy do momentu, aż nie możemy znaleźć punktu „wyższego”



Wypełnianie wieloboku

Zasada parzystości: Prosta, która nie przechodzi przez wierzchołek przecina wielobok parzystą ilość razy.

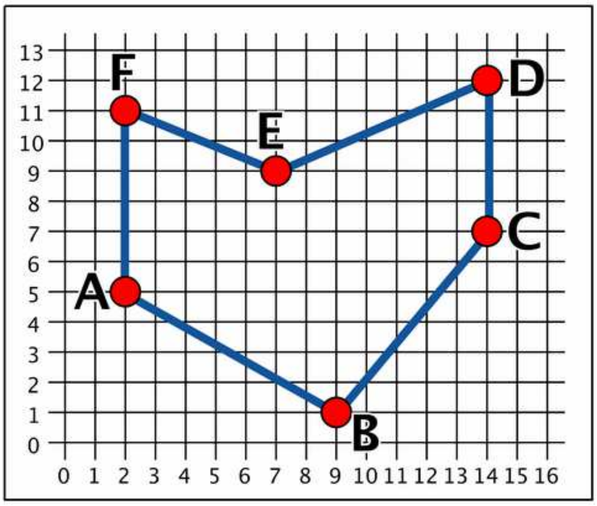


Położenie 1 – 3 przecięcia
Położenie 2 – 6 przecięć
Położenie 3 – 6 przecięć
Położenie 4 - ? przecięć

Algorytm przeglądania linii (scan line)

- Założenie:
  - Dany jest wielobok (bez krawędzi poziomych) opisany jako zbiór kolejnych krawędzi

krawędź	y_{min}	y_{max}	$x(y_{min})$	$x(y_{max})$
AB	1	5	9	2
BC	1	7	9	14
CD	7	12	14	14
DE	9	12	7	14
EF	9	11	7	2
FA	5	11	2	2



Zasada działania algorytmu: Przesuwać poziomą linię skanującą od dołu do góry. Dla danego położenia linii należy rysować punkty pomiędzy poszczególnymi parami krawędzi

Krok 0: Utworzyć globalną tablicę krawędzi (GT)

Krok 1: Ustawić y na najmniejszej wartości współrzędnej y z globalnej tablicy krawędzi (GT), czyli **y** dla pierwszej niepustej grupy krawędzi

Krok 2: Wyzerować aktywną tablicę krawędzi (AT)

Krok 3: Powtarzać tak długo, dopóki tablica globalna (GT) i tablica aktywna (AT) nie będą puste:

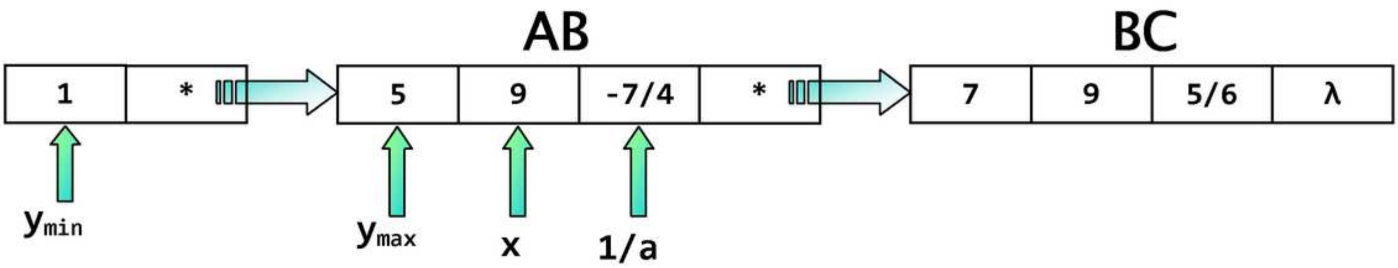
- Przenieść z grupy **y** tablicy globalnej (GT) do tablicy aktywnej (AT) te krawędzie, dla których $\mathbf{y}_{\mathtt{min}} = \mathbf{y}$ i posortować je ze względu na **x**
- Wypełnić piksele w linii y, wykorzystując pary x z tablicy aktywnej (AT)
- Usunąć z tablicy aktywnej (AT) te krawędzie, dla których $\mathbf{y} = \mathbf{y}_{\mathtt{max}}$
- Zwiększyć y o **1** (przejście do następnej linii)
- Dla każdej pary krawędzi, która nie jest pionowa wyliczyć i wstawić do tablicy aktywnej (AT) nowe wartości **x**

Przykład:

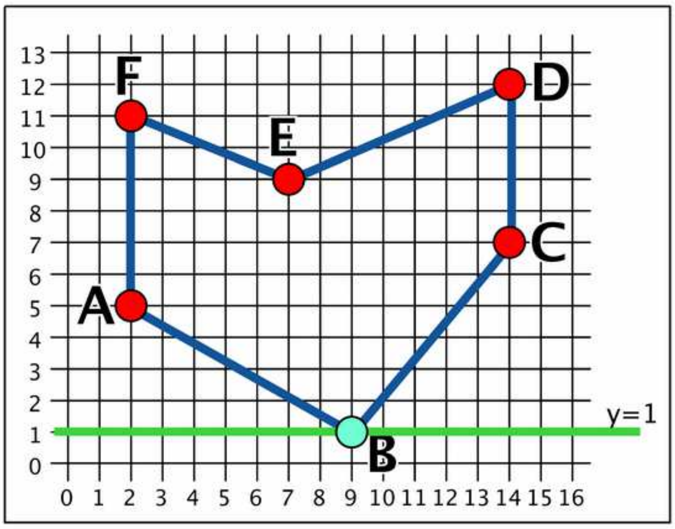
Krok 1: Ustawiamy **y** na najmniejszej wartości współrzędnej y z globalnej tablicy krawędzi **y=1**

Krok 2: Zerujemy aktywną tablicę krawędzi (AT)

Krok 3: Przenosimy z grupy tablicy globalnej (GT) do tablicy aktywnej (AT) te krawędzie, dla których $\mathbf{y}_{\mathtt{min}} = \mathbf{y}$ i sortujemy krawędzie ze względu na **x**



Wypełniamy piksele w linii y, wykorzystując pary x z tablicy aktywnej (AT) i zasadę parzystości:



Usuwamy z tablicy aktywnej (AT) te krawędzie, dla których $y = y_{max}$ (brak takich krawędzi)

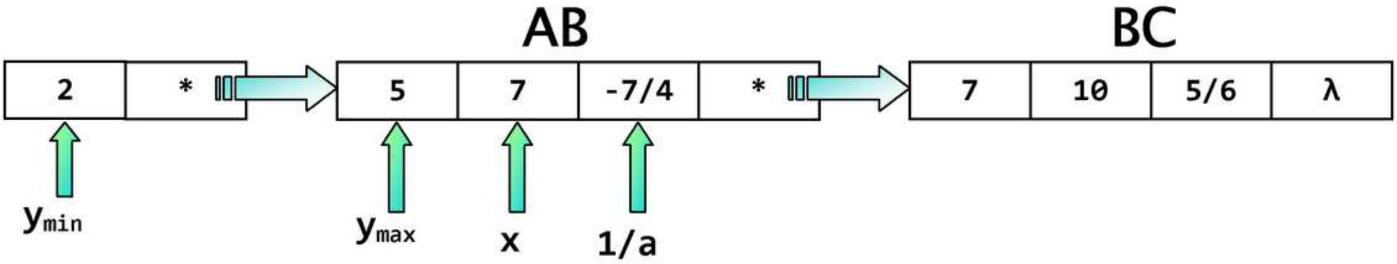
Zwiększamy y o 1 (y=2)

Dla każdej pary krawędzi, która nie jest pionowa wyliczamy i wstawiamy do tablicy aktywnej (AT) nowe wartości x

Sposób obliczania nowych wartości x:

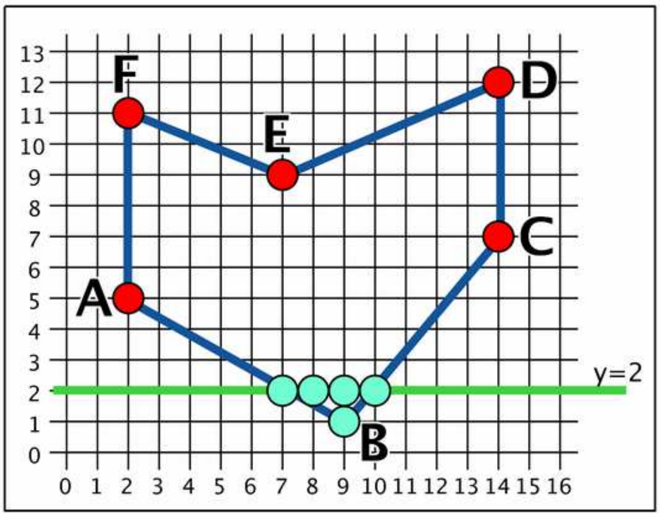
a) Algorytm DDA $\quad x(y) = x(y-1) + \frac{1}{a}$

b) Algorytm Bresenhama



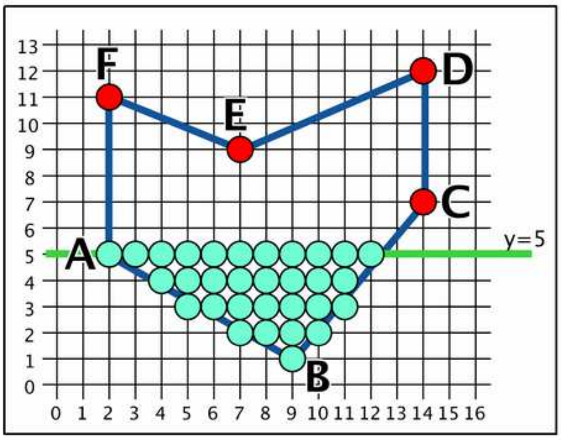
Koniec pierwszego przebiegu dla kroku 3

W kolejnym przebiegu kroku 3 uzyskamy:



Postępujemy analogicznie aż do osiągnięcia y=5.

Dla y=5 mamy:



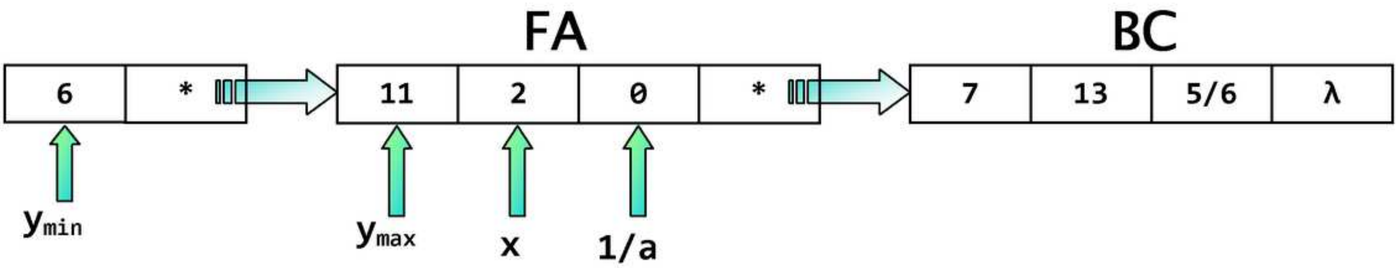
Dla krawędzi AB zachodzi $y = y_{max}$ (koniec krawędzi)

Usuwamy AB z tablicy aktywnej (AT)

Przenosimy do tablicy aktywnej (AT) z tablicy globalnej (GT) krawędź FA, dla której $y_{min} = y$

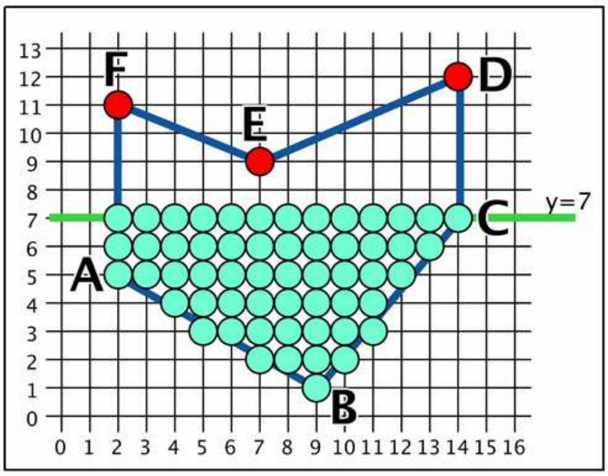
Porządkujemy tablicę aktywną (AT) ze względu na x

Tablica aktywna przybiera postać:

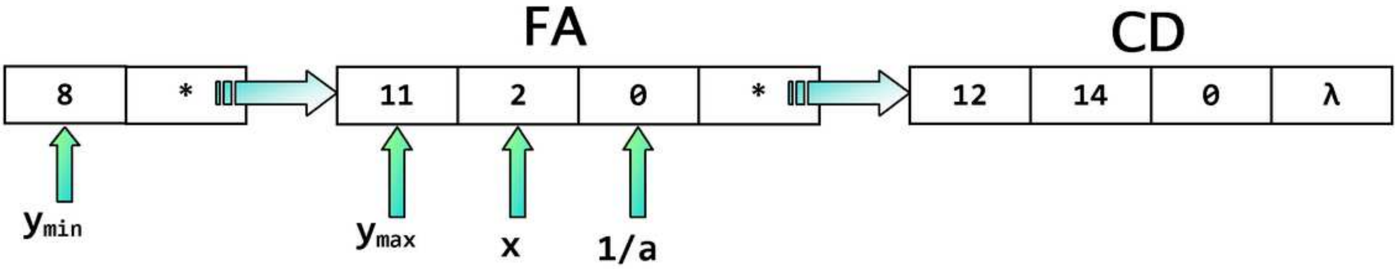


Kontynuujemy wypełnianie aż osiągniemy koniec krawędzi występującej w tablicy aktywnej (AT), lub początek krawędzi z tablicy globalnej (GT).

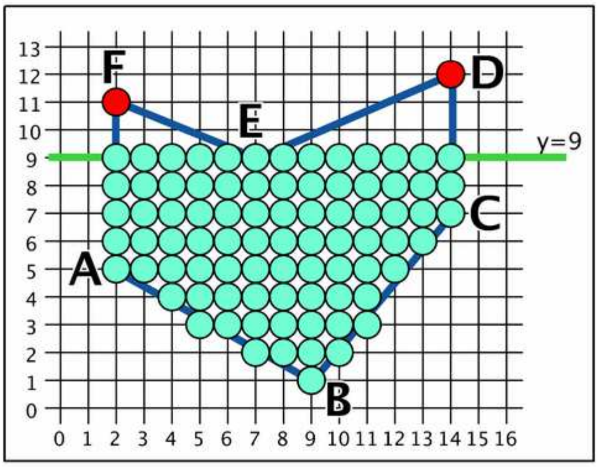
W rozważanym przypadku dla y=7 osiągamy koniec krawędzi BC.



Po usunięciu z tablicy aktywnej (AT) krawędzi BC, zwiększeniu y i przeniesieniu z tablicy globalnej (GT) krawędzi CD tablica aktywna przybiera postać:

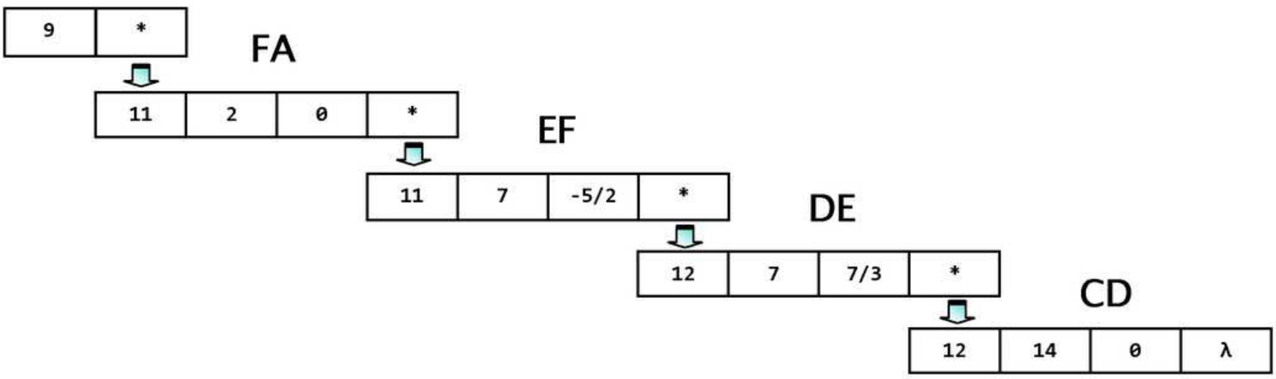


Wypełniając dalej przy y=9 mamy:



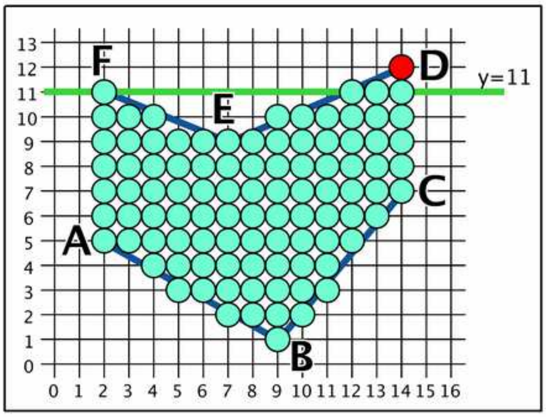
Dla y=9 para krawędzi EF i DE spełnia warunek $y_{min} = y$

W związku z tym przenosimy krawędzie EF i DE do tablicy aktywnej (AT), która po uporządkowaniu elementów ze względu na x przyjmuje postać:

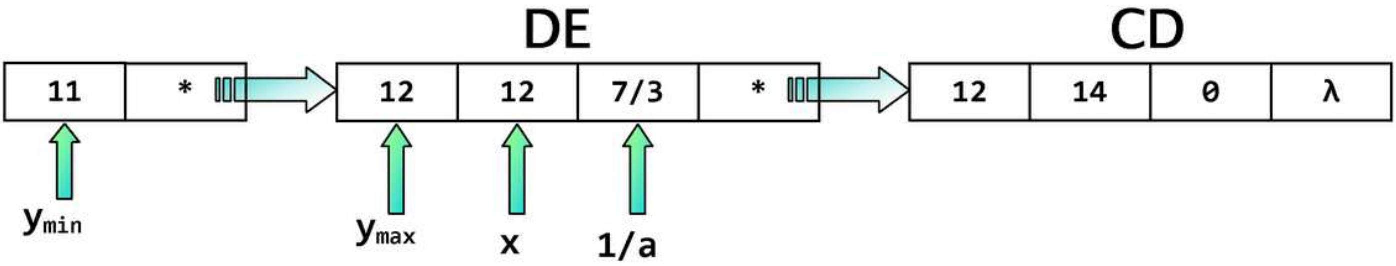


Tablica globalna (GT) staje się pusta.

Kontynuujemy wypełnianie między parami krawędzi z tablicy (AT), aż do osiągnięcia y=11.



Usuwamy z tablicy (AT) parę krawędzi FA, EF:

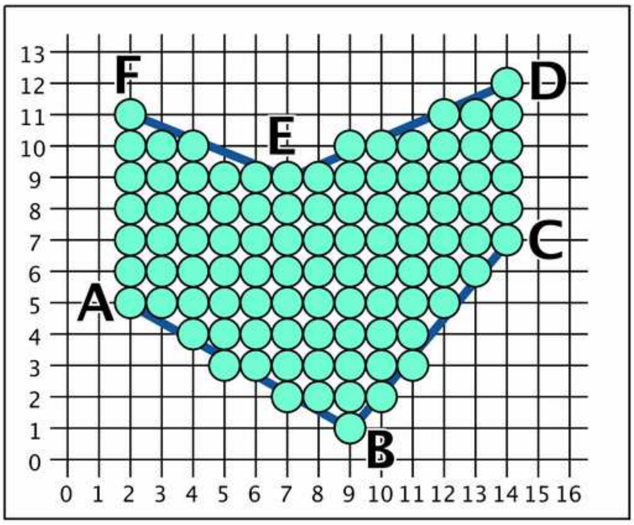


Wypełniamy następną linię i osiągamy końce krawędzi DE i CD.

Usuwamy krawędzie z tablicy AT.

Tablica (AT) jest pusta, czyli kończymy wypełnianie.

Ostateczny efekt:



Problem krawędzi poziomych

Przyjęto założenie, że wypełniany wielobok nie ma krawędzi poziomych. Jak pozbyć się tego założenia?

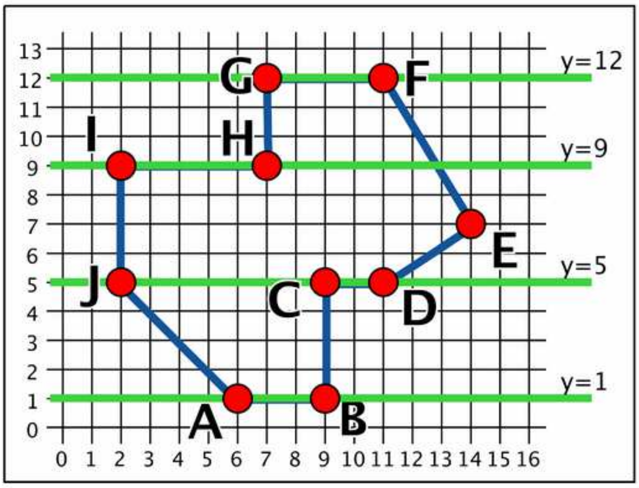
Wystarczy pomijać krawędzie poziome w tablicy (AT)

Dla y=1 w (AT) będą AH i BC

Dla y=5 w (AT) będą IJ i DE

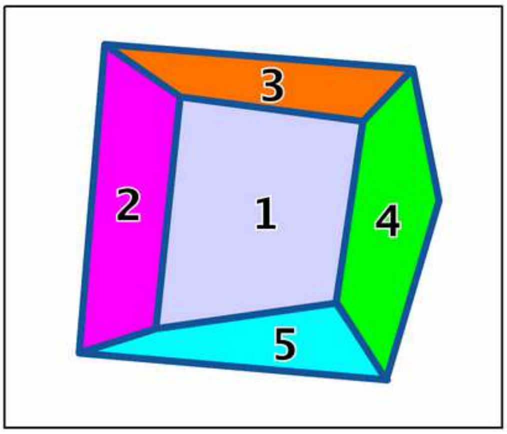
Dla y=9 w (AT) będą IJ i EF

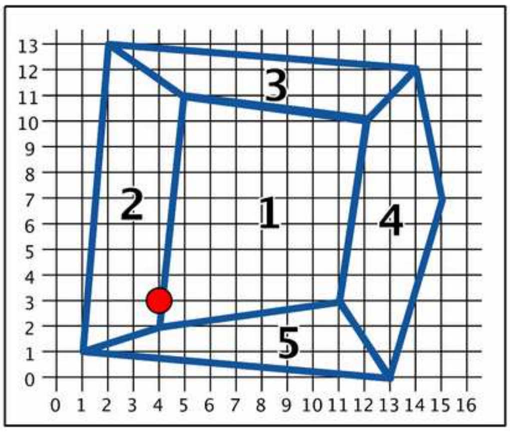
Dla y=12 w (AT) będą GH i FE



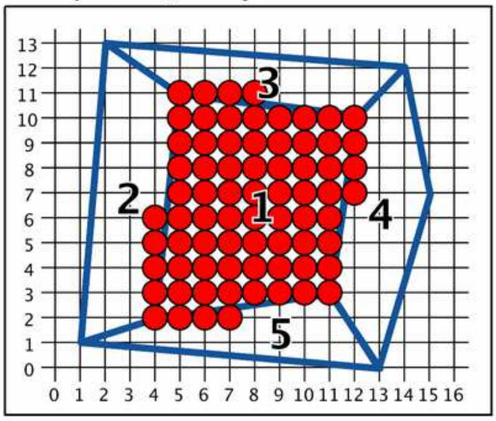
Problemy wypełniania

Problem brzegu wieloboku: Należy narysować kilka wypełnionych wieloboków o wspólnych krawędziach. Każdy z wieloboków wypełniony innym kolorem. Jak rysować obrazy krawędzi wieloboków?

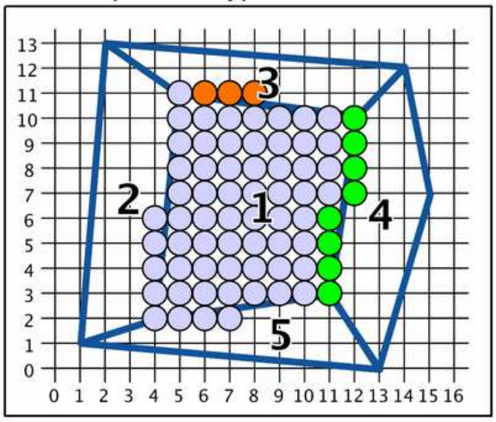




piksele „kandydaci”



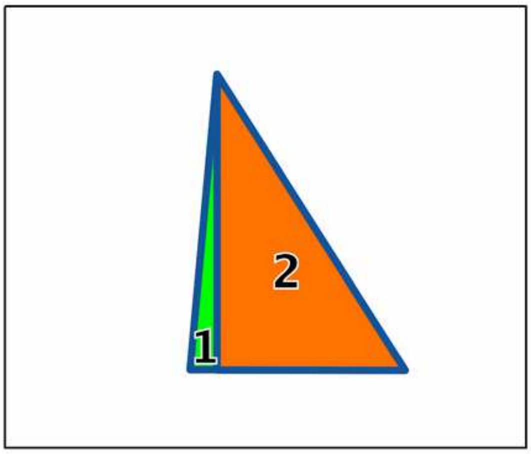
piksele wypełnione

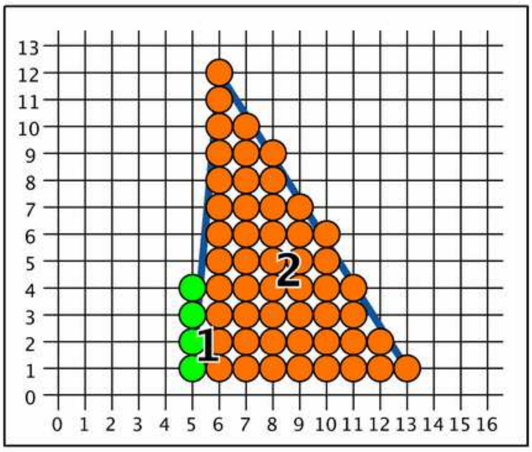


Często stosowane rozwiązanie:

- Rysować piksele leżące wewnątrz wieloboku, ale nie na brzegu
- Rysować piksele należące do lewej krawędzi
- Rysować piksele należące do dolnej krawędzi

Problem wieloboków „bardzo wąskich”: Należy narysować kilka wypełnionych wieloboków o wspólnych krawędziach. Każdy z wieloboków wypełniony innym kolorem. Jak rysować obrazy krawędzi wieloboków?

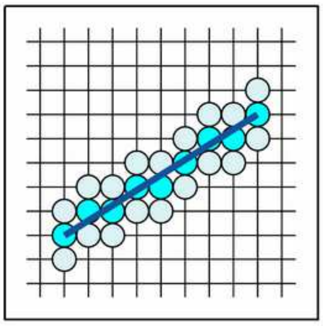




- Zastosowano poprzednio opisaną konwencję rysowania.
- Obraz wieloboku 1 składa się tylko z czterech punktów.
- Brak zadawalającego rozwiązania przy tym sposobie rysowania.
- Należy zastosować wypełnianie wielotonowe.

Rysowanie pogrubionych prymitywów

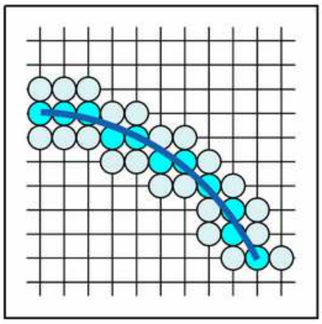
Metoda powielania pikseli



Dla każdego piksela rysunku podstawowego, rysowane są dodatkowe piksele pogrubiające:

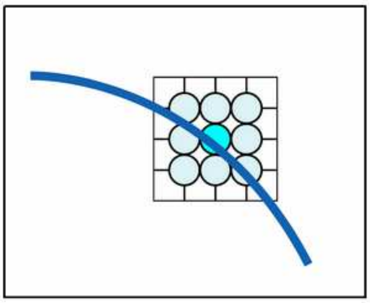
- Gdy **|a|<1** w kolumnie,
- Gdy **|a|≥1** w wierszu

Metoda powielania pikseli dla krzywej



Zaleta: prosty algorytm
Wada: grubość rysunku dla różnych fragmentów krzywej jest różna

Metoda prostokątnego pióra



- „ślad" pióra porusza się wzdłuż krzywej
- piksel rysunku podstawowego znajduje się w centrum śladu
  - Zaleta: lepsze niż w poprzedniej metodzie rysunki
  - Wada: występuje powielanie pikseli