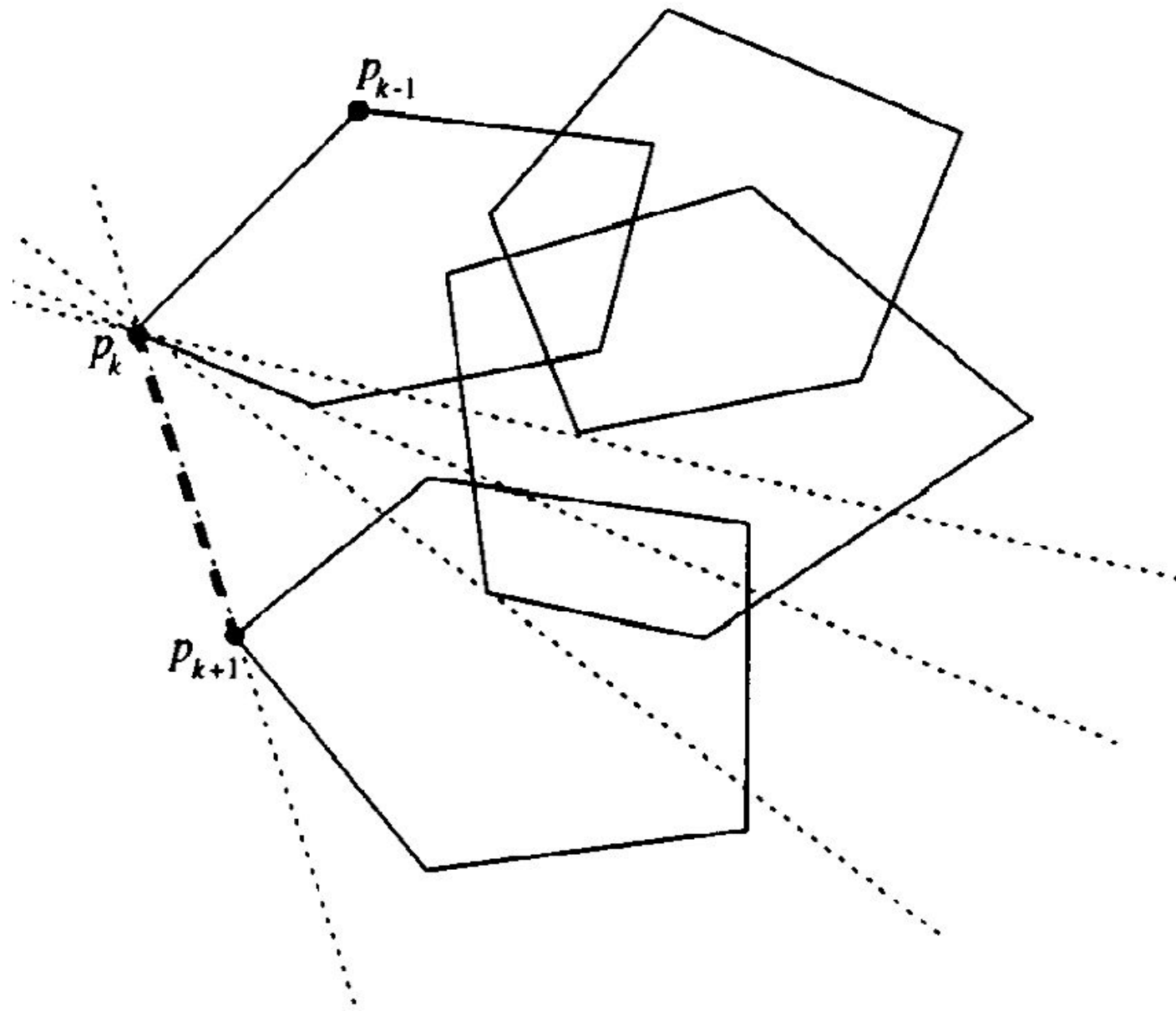
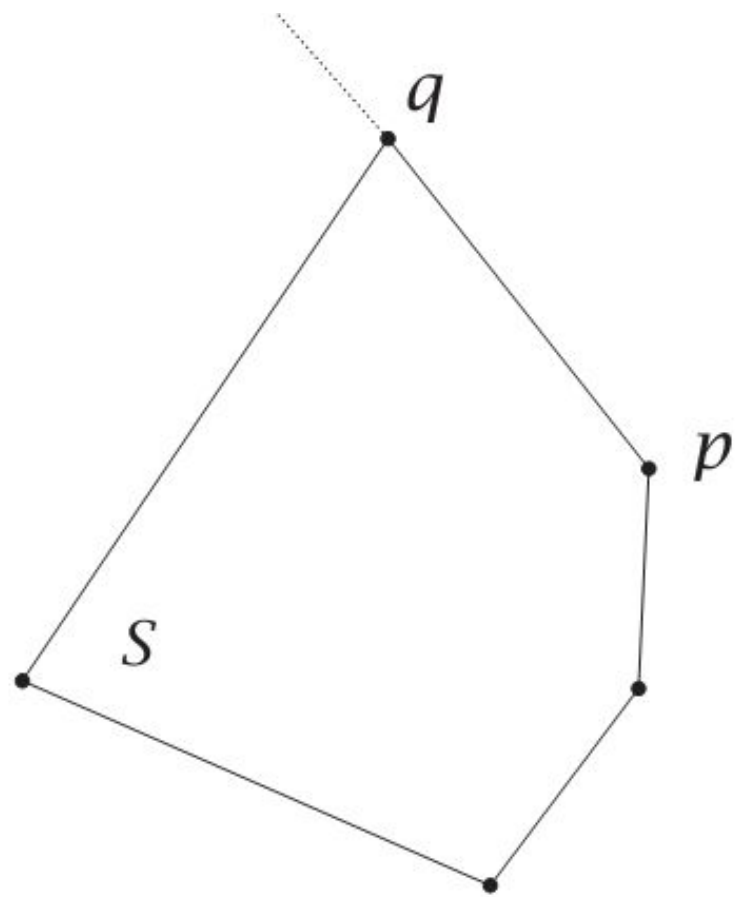
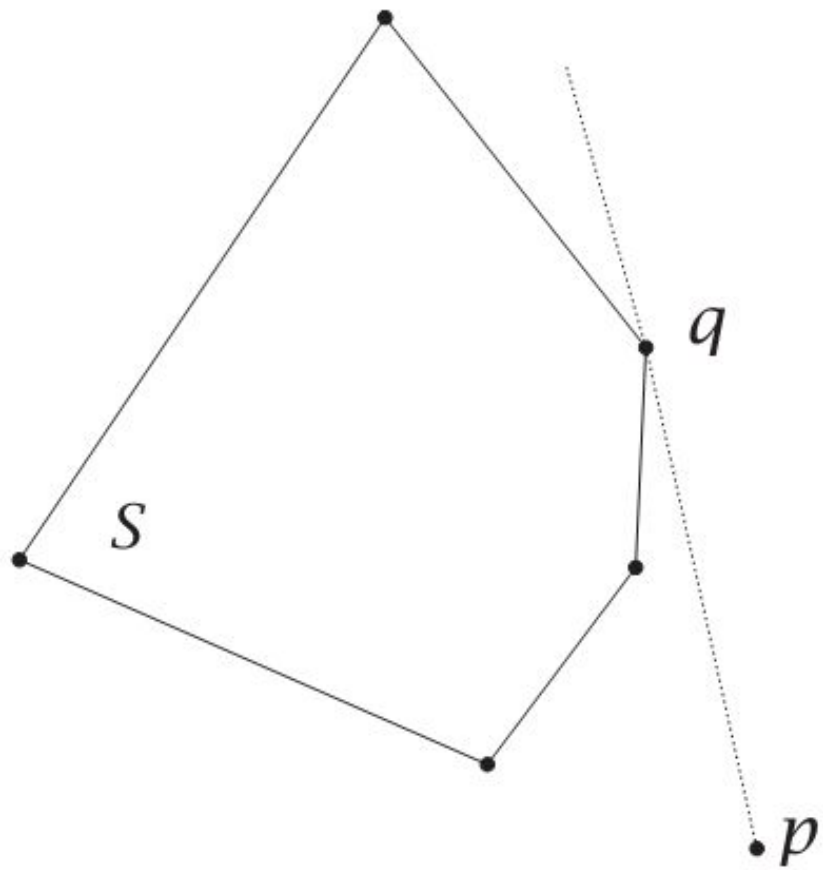


# Алгоритм Чена (1996)





- 1 Розділимо множину  $P$  на  $n/m$  непересічних підмножин  $P_i$
- 2 Побудуємо опуклі оболонки  $CH(P_i)$
- 3 Знайдемо точку  $p_{start}$ , яка буде гарантовано включена в опуклу оболонку  $CH(P)$
- 4 Будемо виконувати кроки, знаходячи кожного разу таку точку, яка є наступною вершиною опуклої оболонки в порядку обходу проти годинникової стрілки
- 5 Коли чергова знайдена точка співпадає з  $p_{start}$  будемо вважати, що опукла оболонка  $CH(P)$  побудована



$$O(n \log m + h \lceil \frac{n}{m} \rceil)$$

**for** t = 1; 2; 3;... **do**

  M := min (n, 2^(2^t))

  Викликати модифікацію Chan (P; m)

**if** Алгоритм побудував опуклу оболонку CH (P) **then**

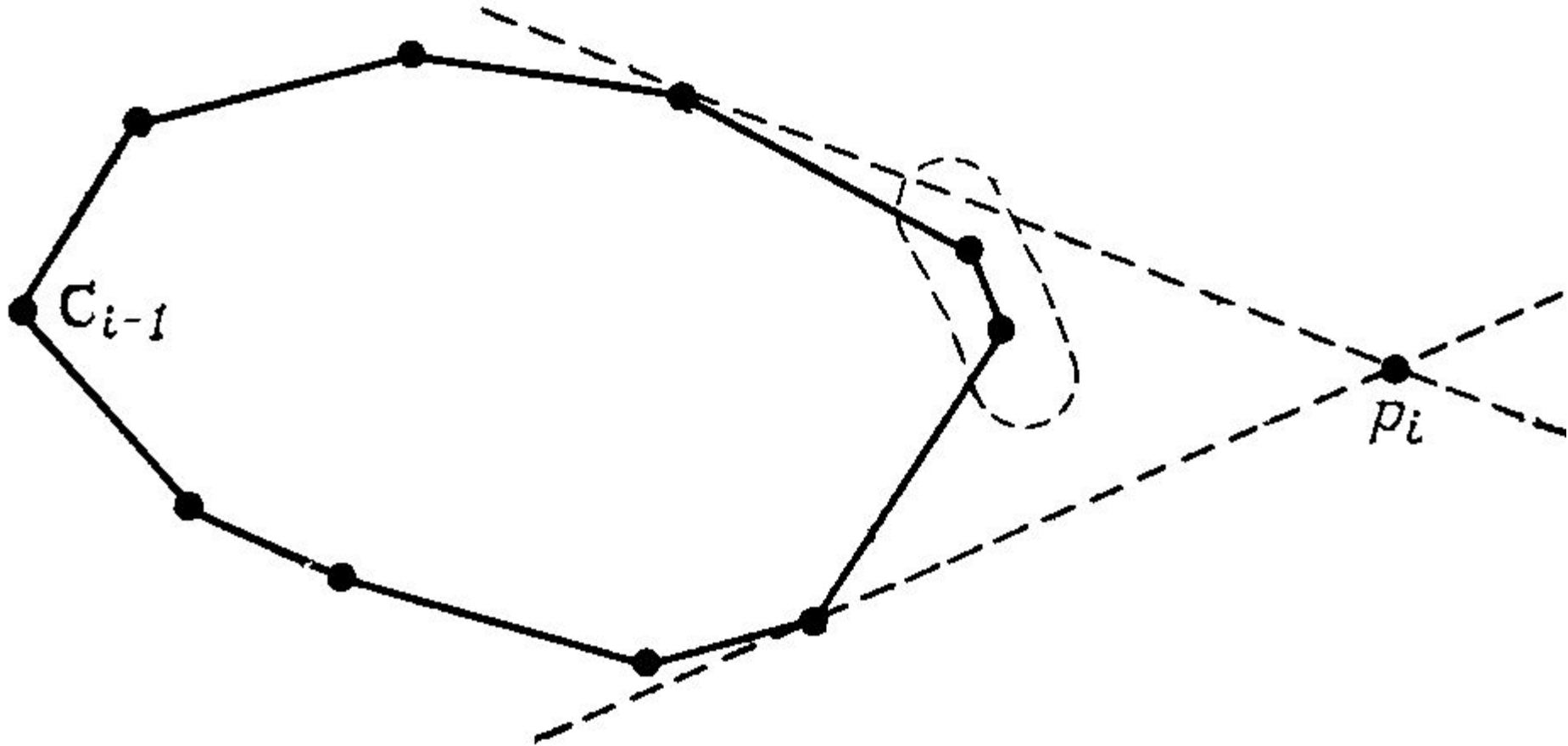
    Повернути в якості результату CH (P)

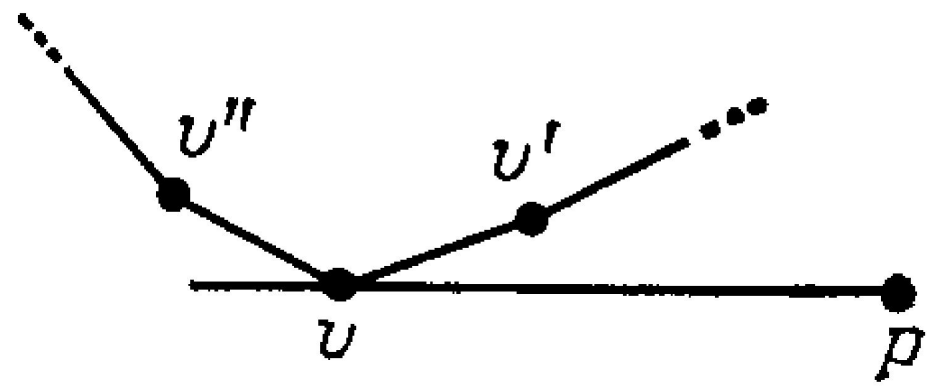
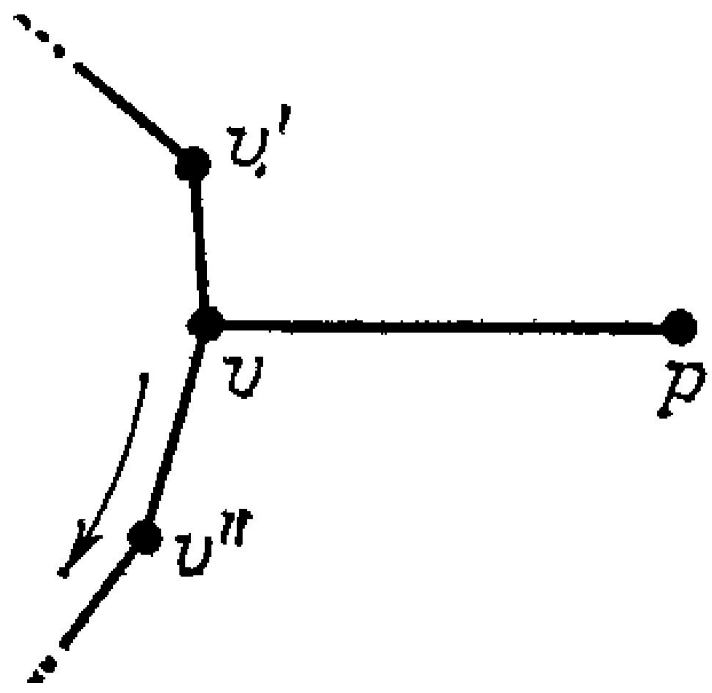
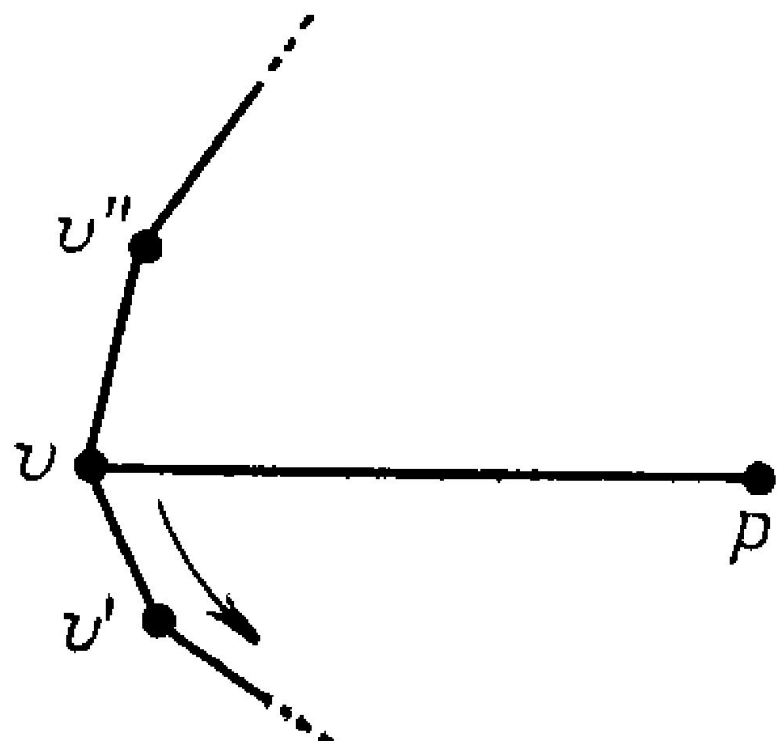
**end-then**

**end-do**

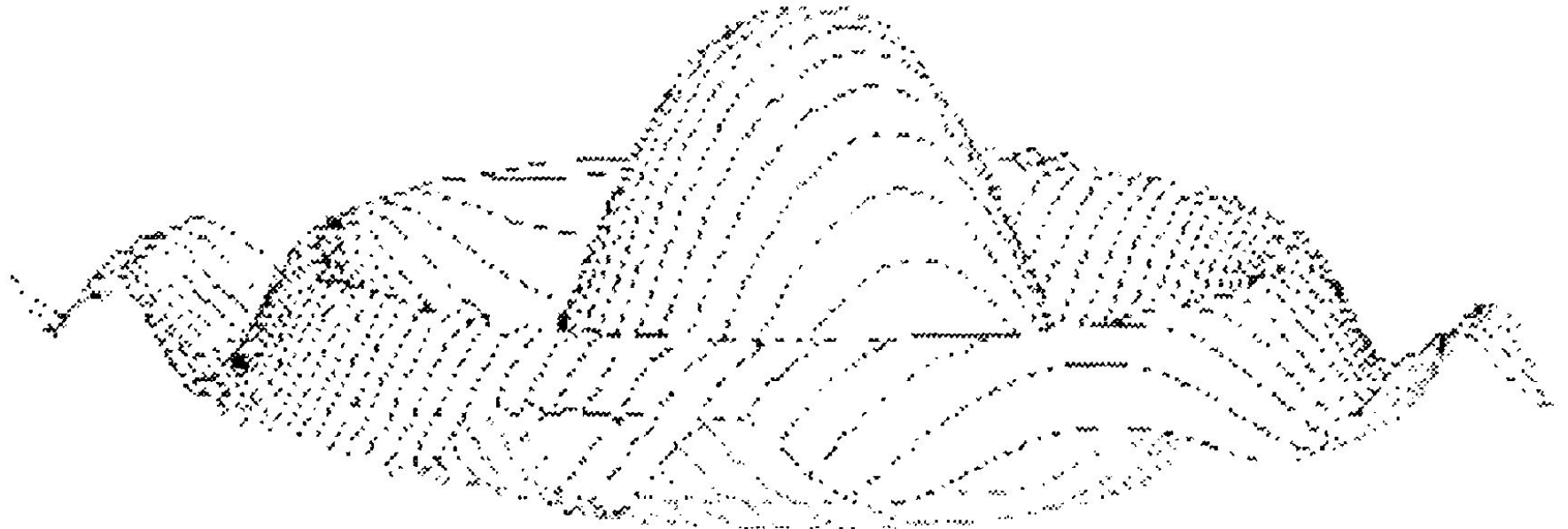
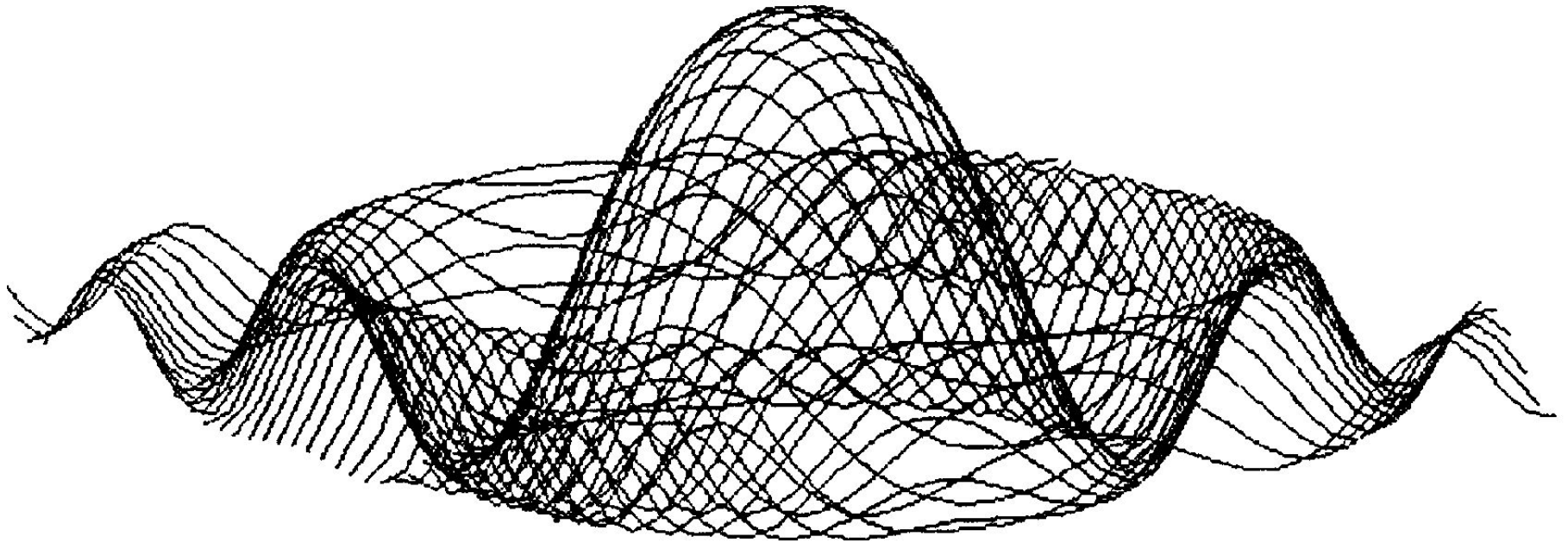
$$O\left(\sum_{i=1}^{\lceil \log \log h \rceil} n 2^i\right) = O(n 2^{\lceil \log \log h \rceil + 1}) = O(n \log h)$$

# Побудова ОБ в реальному часі

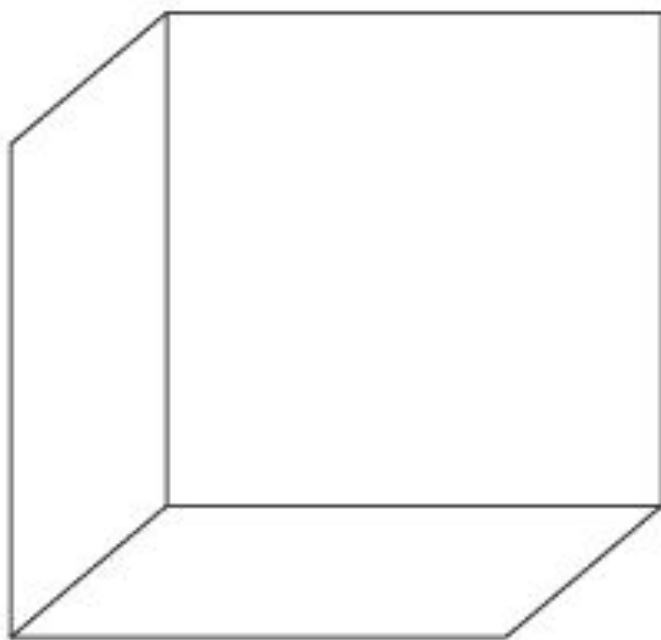
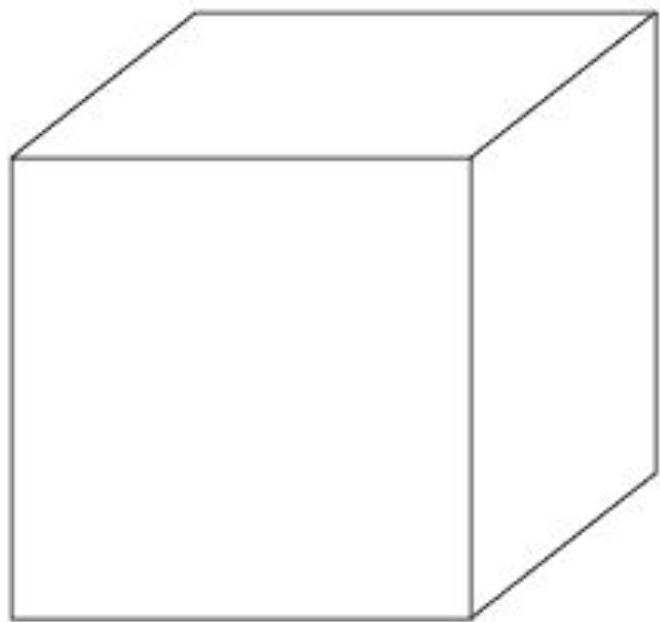
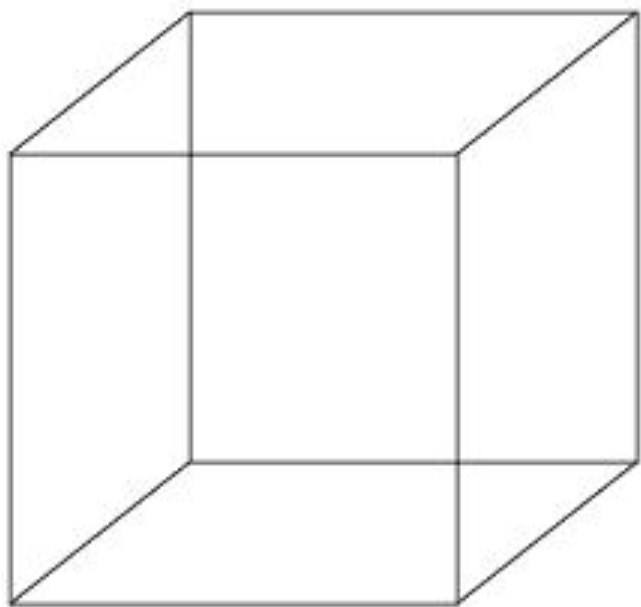




# ВИДАЛЕННЯ НЕВИДИМИХ ГРАНЕЙ, РЕБЕР ТА ВЕРШИН







# ПАМЯТКА ЗАКАЗЧИКУ

Ты сам веришь, что такое бывает?..



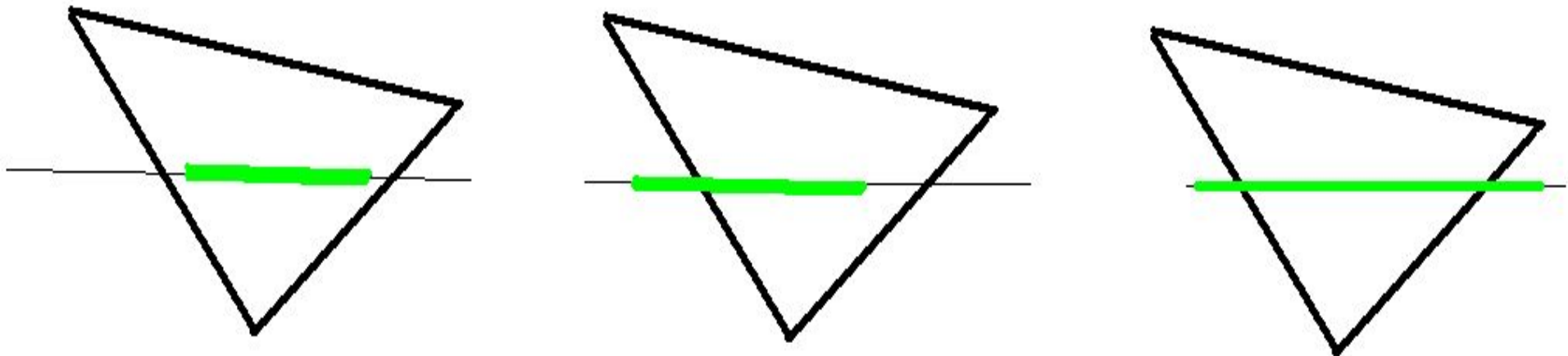


Алгоритми об'єктних методів працюють з об'єктними координатами примітивів і точок.

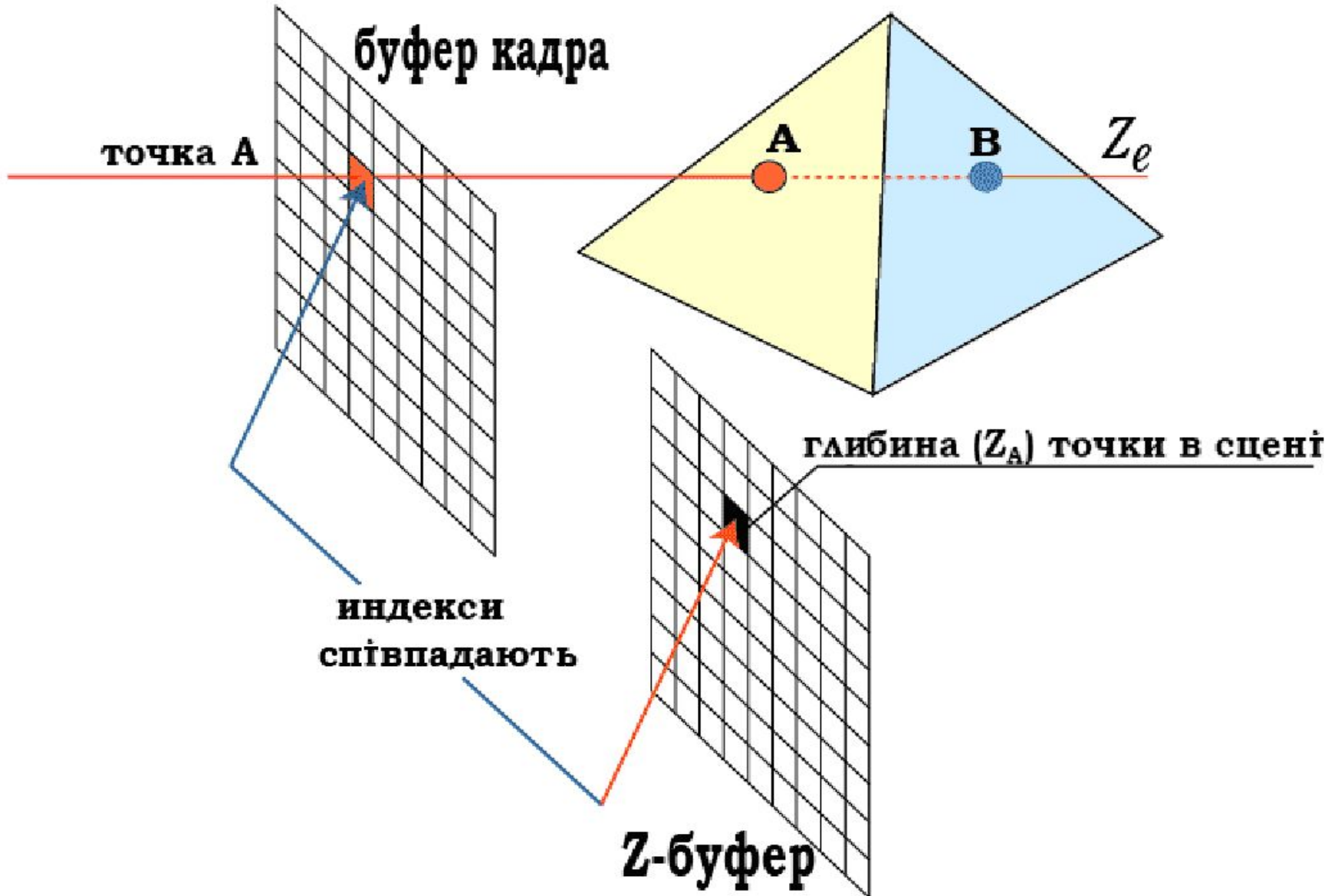
Алгоритми екранних методів працюють з координатами пікселів, які зображують на екрані точки сцени.

# Алгоритм Робертса

- Відкидаються ребра, що належать не лицьовим граням
- Кожне з ребер перевіряється на закривання лицьовими гранями:
  - Ті ребра, що повністю вкриваються – відкидаються
  - Частково вкриті ребра скорочуються або розбиваються на два



# Z-буфер



Для кожного пікселя  $[x, y]$  буфера кадру

Begin

If  $Z[x, y] < zbuf[x, y]$  then begin

Колір  $[x, y]: =$  КолірТочкиСцени  $[x, y];$

$zbuf[x, y]: = Z[x, y];$

end;

End;

# Ієрархічний Z-буфер

10101010	7	101010
101010	6	6 6 1010
1010	5 5	6 6 1 10
10	4 4 4	5 5 5 10
10	4 4 4	3 3 4 4
101010	1 1 1	3 3
101010	1	10101010
1010101010101010		

10101010		
10	5	6 10
1010	3	4
10101010		

1010
1010

10
----