**Звіт**

**Студента 3 курсу**

**Групи ІС-3**

**Пригоди Андрія**

**Задача:**

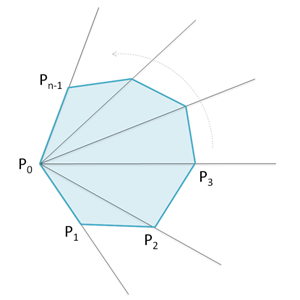
**Приналежність точки опуклому многокутнику.**Існує декілька методів вирішення цієї задачі, починаючи від перевірки на парність кількості перетинів променя, випущеного з точки, з ребрами багатокутника і закінчуючи двійковим пошуком.

У першому випадку складність буде O (n), n - кількість вершин багатокутника, а в другому - O (log2n).

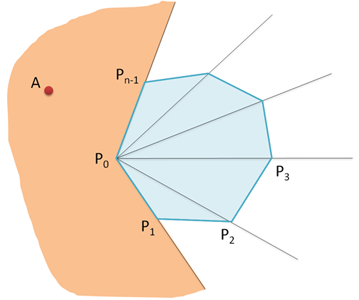
Розглянуто буде другий метод вирішення, а саме шляхом двійкового пошуку.

**Двійковий пошук**

Маємо опуклий многокутник P, що складається з n вершин та точку A.



Сутність алгоритму: беремо першу вершину многокутника P0 та намагаємося визначити, в який сектор (кут) PiP0Pi+1 потрапляє точка A. Спочатку перевіряємо, чи точка A потрапляє в сектор Pn-1P0P1. Якщо це не так, то A гарантовано лежить поза многокутником.



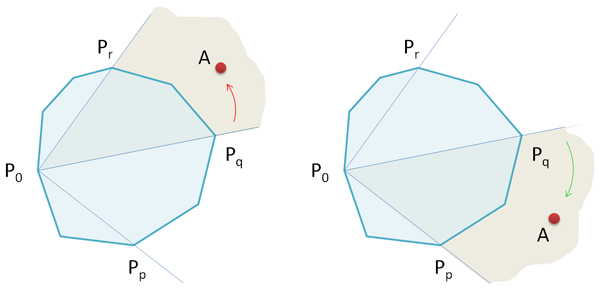
Код: (на python)  
def pointloc(P,A):

n = len(P)

if rotate(P[0],P[1],A)<0 or rotate(P[0],P[n-1],A)>0:

return False

Далі вважаємо p = 1, r = n - 1 (межі поточного сектора), вимірюємо середню вершину q = (p + r)/2. Дивимось, де знаходиться A відносно вектора P0Pq. Якщо зліва, то замінюємо r на q. Якщо зправа, то замінюємо p на q.



Продовжуємо, доки не виявимо, що r – p = 1:

p, r = 1, n-1

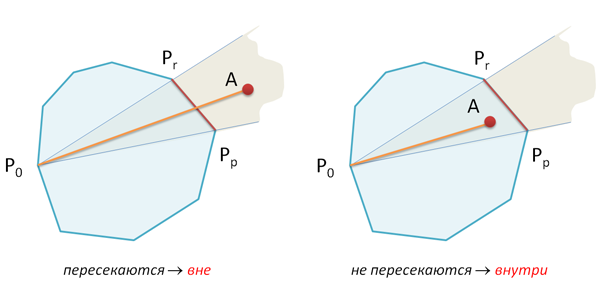
while r-p>1:

q = (p+r)/2

if rotate(P[0],P[q],A)<0: r = q

else: p = q

Залишилось перевірити, чи перетинаються відрізки P0A і PpPr?   
Якщо перетинаються, то точка A лежить поза. Якщо не перетинаються, то належить.



Код:

return not intersect(P[0],A,P[p],P[r])

**Використане джерело:** <https://habrahabr.ru/post/144571/>