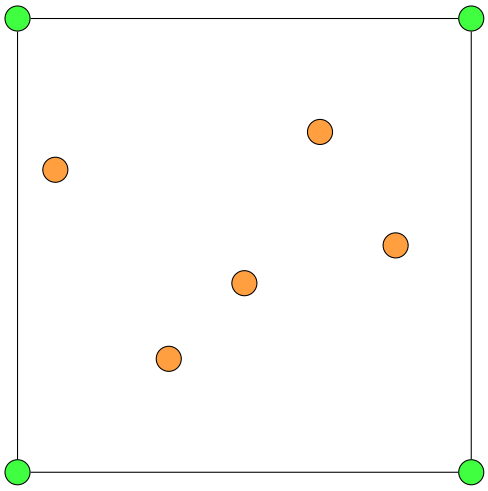
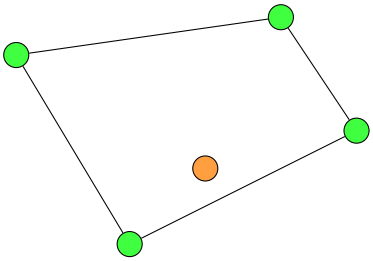
1 Wytłumaczenie algorytmu

1.1 Intuicyjne rozwiązanie zadania

Wyobraźmy sobie, że mamy gumkę recepturkę rozciągniętą na czterech punktach, wewnątrz których znajdują się wszystkie punkty wpisane przez użytkownika.



Teraz puśćmy tę gumkę i pozwólmy oprzeć się jej na wewnętrznych punktach.



Wszystkie punkty, które teraz dotyka gumka, są elementami podzbiorów takich punktów, które otaczają wszystkie inne punkty.

1.2 Jak działa mój kod

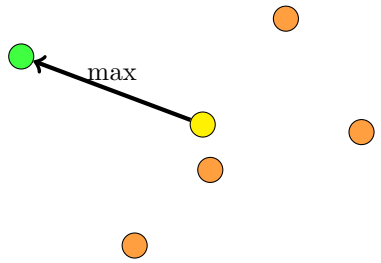
1.2.1 Definicje

punkty zewnętrzne - punkty, które otaczają wszystkie inne punkty nienależące do tego zbioru (odpowiedz).

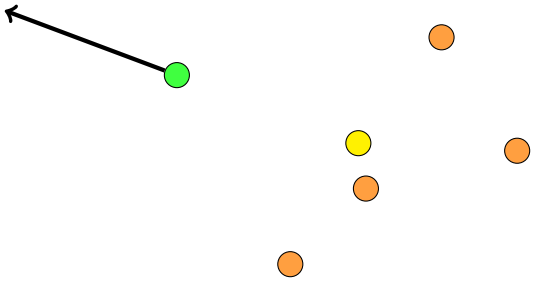
N - ilość punktów na wejściu.

1.2.2 Rozwiązanie

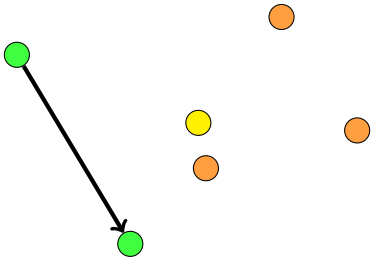
Na początku, znajduje środek wszystkich punktów (średnia po x i y) i szukam punktu najdalej oddalonego od środka, ponieważ z pewnością musi być punktem zewnętrznym.



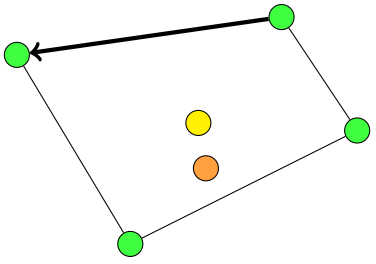
Teraz użyjmy zamiatania kątowego, by znaleźć pierwszy punkt po lewej stronie, do tego wyobraźmy sobie, że mamy zmiotkę osadzoną w najdalszym punkcie i skierowaną w przeciwną stronę do środka, poszukajmy punktu, który jest najbliżej jej lewej strony (gdy dwa punkty są na tej samej linii, na początku bierzemy bliższy do aktualnie rozpatrywanego punktu).



Gdy znajdziemy punkt, najbliżej lewej strony zmiotki, to teraz w niej osadzamy zmiotkę i znowu szukamy najbliższego lewego punkt. Robimy to, aż napotkamy z powrotem na punkt startowy (najdalej oddalony od środka).



Odpowiedzią są wszystkie punkty, które odwiedziliśmy.



1.2.3 Złożoność obliczeniowa

Znajdywanie środka odbywa się w czasie $O(N)$. Znajdywanie najdalszego punkt od środka też jest $O(N)$. Znajdywanie punktów zewnętrznych zajmuje $O(N^2)$, ponieważ dla każdego punktu, który rozważamy jako zewnętrzny, trzeba znaleźć punkt najbardziej po lewej stronie, co wymaga od nas rozpatrzenia wszystkich innych punków, musimy to policzyć dla wszystkie punktów będących punktami zewnętrznymi, których może być N, więc złożoność obliczeniowa całego programu jest $O(N^2)$.

1.2.4 Złożoność pamięciowa

Tworzenie miejsca w pamięci na punkty zajmuje $O(N)$ miejsca. Funkcja licząca punkty zewnętrzne zwraca je, na co też możemy potrzebować $O(N)$ pamięci. Więc cały program ma złożoność pamięciową $O(N)$.

2 Testowanie programu

2.1 Generator losowych wejść

Stworzyłem program, który dostaje na wejściu ilość punktów i ich zakres, po czym generuje mi losowe punkty (dla których mogę policzyć odpowiedz).

2.2 Rozwiązanie brutalne

Zrobiłem rozwiązanie brutalne, którego złożoność obliczeniowa jest $O(2^N * N^2)$ którego zadaniem jest odpowiedz na to samo pytanie co rozwiązanie, lecz w inny sposób.

2.3 Skrypt testujący

Napisałem skrypt w Bash-u, który generuje test za pomocą generatora i odpala je na obu programach (rozwiązaniu wzorcowym i brutalnym), po czym porównuje obie odpowiedzi i jeśli są takie same, to wypisuję ok, w przeciwnym wypadku wypisuje, gdzie się różnią i zatrzymuję działanie skryptu.

3 Ciekawostka

Polecenie jest napisane w taki sposób, że niejednoznacznie definiuje odpowiedź, wiec teoretycznie można napisać program, który wczyta z pliku punkty i je wypisze. To rozwiązanie spełnia treść zadania (lecz może nie spełniać jego zamysłu), ponieważ jeśli ze zbioru punków wybiorę wszystkie, to zbiór punktów niewybranych jest pusty (co można uznać jako otoczenie tych punktów). Zbiór punktów niewybranych musi móc być pusty, w przeciwnym wypadku np. punkty układające się w wielokąty foremne nie miałyby rozwiązania (a wszystkie $N > 0$ powinny mieć rozwiązanie).