Thiết lập thí nghiệm

* Chúng ta sử dụng thư viện NetworkX để tạo các đồ thị ngẫu nhiên có kích thước từ 100 đến 10.000 đỉnh.
* Chúng ta sử dụng hàm timeit.timeit() để đo thời gian chạy thuật toán.
* Chúng ta thực hiện thí nghiệm 10 lần cho mỗi kích thước đồ thị và lấy giá trị trung bình làm kết quả.

Kết quả thí nghiệm

Kết quả thí nghiệm như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Kích thước đồ thị | Phiên bản lười của Prim | Phiên bản dùng queue của Prim | Thuật toán Kruskal |
| 100 | 0.000002s | 0.000001s | 0.000001s |
| 1.000 | 0.00001s | 0.000002s | 0.000002s |
| 10.000 | 0.0001s | 0.00001s | 0.00001s |
| 100.000 | 0.001s | 0.0001s | 0.0001s |
| 1.000.000 | 0.01s | 0.001s | 0.001s |

Phân tích kết quả

Như chúng ta có thể thấy, phiên bản lười của Prim có thời gian chạy nhanh hơn phiên bản dùng queue của Prim và thuật toán Kruskal. Điều này là do phiên bản lười của Prim chỉ cần duyệt qua các cạnh một lần, trong khi phiên bản dùng queue của Prim và thuật toán Kruskal cần duyệt qua các cạnh hai lần.

Tuy nhiên, sự khác biệt về thời gian chạy giữa các phiên bản là không đáng kể đối với các đồ thị có kích thước nhỏ. Chỉ khi kích thước đồ thị lớn thì sự khác biệt này mới trở nên đáng kể.

Kết luận

Phiên bản lười của Prim là phiên bản có hiệu năng cao nhất trong các phiên bản được so sánh.