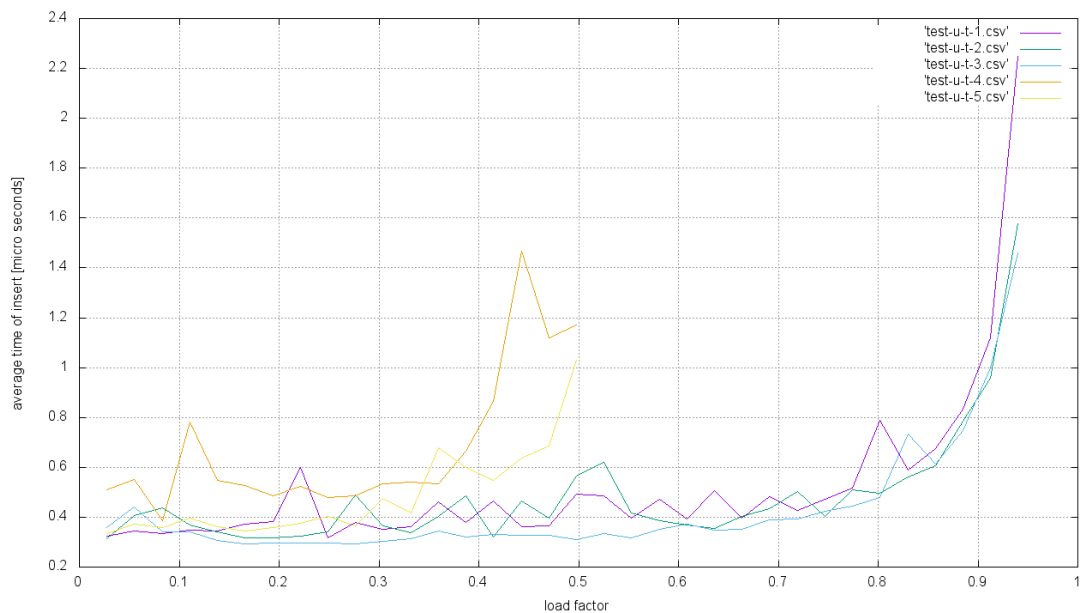


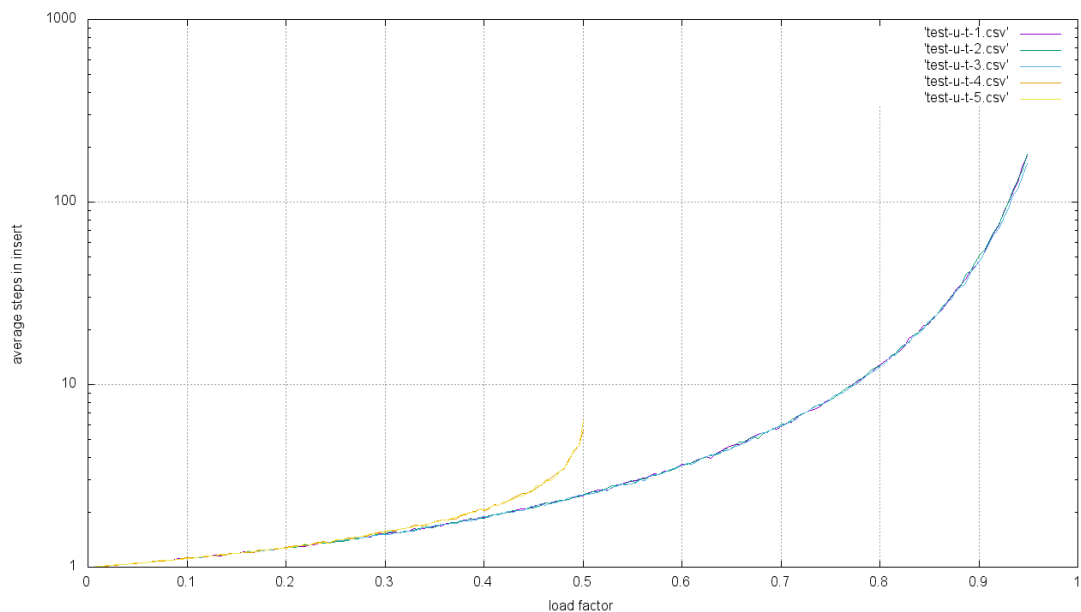
0.1 Náhodný testovací případ

Oba testy (časový i krokový) jsme dělali na tabulkách velikosti 2097151. Pro graf s časy jsme průměrné doby průměrovali přes 58000 insertů, je to hodnota, při které se graf vyčistil na tolik, že bylo možné něco pozorovat. Následující seznam přidává popis k následujícím grafům.

- 1 – linear probing with tabulation hash
- 2 – linear probing with multi hash
- 3 – linear probing with naive mod hash
- 4 – cucko with tabulation hash
- 5 – cucko with mult hash



Z grafu je překvapivě vidět, že naivní hashovací funkce potřebuje většinou průměrně nejmenší dobu na insert. Pravděpodobně je to způsobeno jejím nejjednodušším výpočtem a zároveň přidáváním náhodných prvků z univerza.



V obou grafech také vidíme, že u kukačky začne doba rychle stoupat už u z poloviny plné tabulky. Pro druhý graf ukazující průměrný počet potřebných kroků byly hodnoty průměrovány přes 10000 vložení. Není zde vidět žádný rozdíl plynoucí z rozdílných hashovacích funkcí, což je zřejmě způsobeno tím, že vkládané prvky jsou zcela náhodné.

0.2 Sekvenční testovací případ

Následující seznam je popis k číselnému označení grafů.

- 1 – linear probing with tabulation hash
- 2 – linear probing with mult hash

Na všech následujících grafech je vidět, že hashovací funkce číslo 2 dosahuje lepších výsledků při sekvenčním testu ve skoro plné tabulce.

