

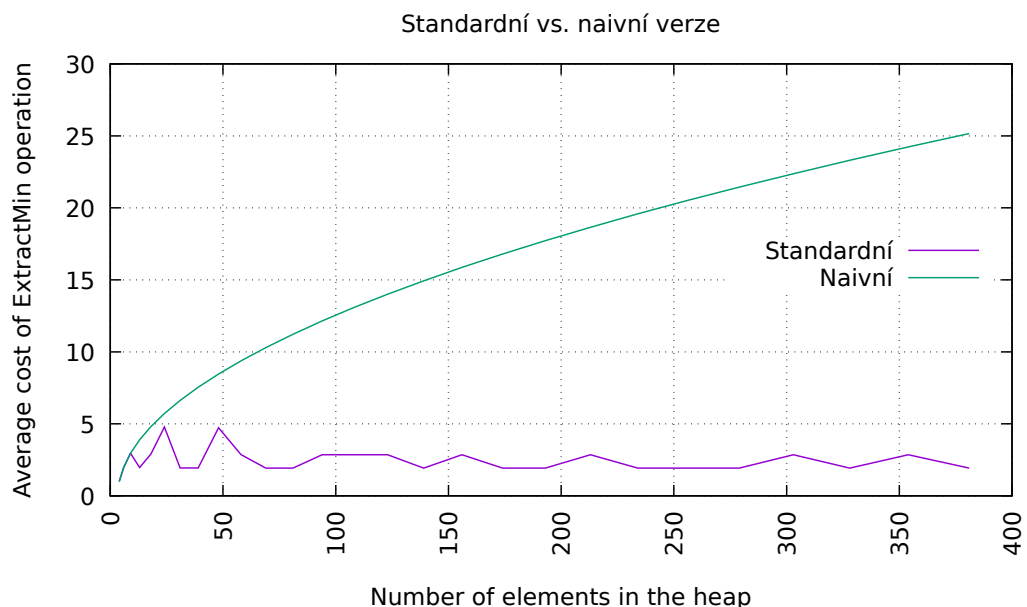
První sada grafů ukazuje počet kroků ExtractMin v rámci dvou různých, náhodných posloupností operací, které se liší pouze podílem operace ExtractMin. Jsou na ní patrné dva zajímavé úkazy.

Z grafu pro rovnoměrnou posloupnost je vidět, že počet kroků s rostoucí velikostí haldy pomalu roste. Z grafu se zdá, že by růst mohl být přibližně logaritmický. To by bylo konsistentní s faktem, že operace ExtractMin má mít amortizovanou logaritmickou časovou složitost a že námi počítané kroky jsou podmnožina operací, které musí ExtractMin provést (nezapočítáváme například stromy z haldového lesa, které v něm byly ještě před odebráním minima a které s žádným jiným v rámci konsolidace nespojujeme). Základ logaritmu je, zřejmě, řádově v nízkých jednotkách.

Z grafu, který obě posloupnosti porovnává, je pak vidět, že podíl operací silně ovlivňuje aditivní konstantu počtu kroků. Kdyby byla ovlivněna převážně multiplikativní konstanta

namísto aditivní, tak by byl pozvolný logaritmický růst patrný i na grafu pro nevyváženou sekvenci. Na něm ovšem není prakticky žádný růst patrný.

Tuto teorii podporuje také nahlédnutí toho, co se s implementací děje při menším podílu operace ExtractMin. Vzhledem k tomu, že konsolidaci a tedy redukci počtu stromů v haldovém lese provádí pouze tato operace, tak její menší četnost způsobí větší podíl jednotkových stromů (které vytváří jak Decrease, tak Insert), které pak musejí být konsolidovány. Počet operací Insert a Decrease mezi každými dvěma ExtractMin nám pak dává, byť extrémně hrubý, odhad na velkou část aditivní konstanty. Z tohoto grafu je také vidět že zatímco multiplikativní konstanta je a zůstává malá, tak aditivní může růst vcelku výrazně. Mezi rovnoměrným a nevyváženým testem vzrostla o několik řádů.



Druhá sada grafů, respektive graf, porovnává naivní implementaci se standardní. Z grafu se zdá, že počet kroků naivní implementace roste přibližně s první odmocninou počtu prvků v haldě.

Tuto hypotézu podporuje i teorie. Pro naivní implementaci totiž, jak mimo jiné říká článek¹ odkazovaný v generátoru, existuje konfigurace haldy a posloupnost operací, která má amortizovanou složitost právě odmocninu z velikosti stromu.

Konkrétně jde o konfiguraci, kdy les haldy obsahuje stromečky v podobě vždy vrcholu s připojeným jedním synem (bez dalších synů), dvěma syny (bez dalších synů), třema syny, a tak dále. Když pak na této haldě vložíme dvě nová minima a následně provedeme dvakrát delete, tak způsobíme odmocnina mnoho konsolidací (všechny stromy, kterých je právě odmocnina je třeba při prvním delete sloučit a při druhém zas vrátit do haldového

¹Replacing Mark Bits with Randomness in Fibonacci Heaps, Jerry Li, John Peebles 2014

lesa) a vrátíme se do původní konfigurace. Můžeme tedy operaci opakovat a být si jisti, že se neuamortizuje. Multiplikativní konstanta je v tomto případě větší než 1, ale není nikterak extrémně velká. Řádově nízké jednotky.