- 1a)

Python: TimSort (AD_08_PraktischeDatenstrukturen.pdf Seite 24)

PHP: Quicksort (https://www.php.net/manual/en/function.sort.php#refsect1-function.sort-notes)

- 1b)

Verwendete Algorithmen: Quicksort, Heapsort Insertionsort.

Introsort fängt mit Quicksort an. Sobald die Rekursionstiefe ein gewisses Level basierend auf der Anzahl der Elemente überschreitet wird das Teilstück mit Heapsort sortiert. Wenn die Anzahl der Elemente eine gewisse (statische) Anzahl unterschreitet wird mit Insertionsort sortiert.

Vorteile: Die Vorteile von allen drei Algorithmen werden kombiniert. Die performance ist (für typische Datensets) mit der von Quicksort vergleichbar und im worst-case O(n log n) (dank Heapsort).

Source: https://en.wikipedia.org/wiki/Introsort

- 1c)

Das Array wird von vorne (Index 0) nach hinten (Index n-1) durchlaufen und Elemente nur dann vertauscht, wenn das vorherige echt größer als das nächste ist. Daher sollte wenn zwei gleiche Elemente auftreten das später auftretende immer nach dem früher auftretendem zu liegen kommen.

- 2a)

Geeignet, da eine simple Division in Polynominalzeit durchführbar ist.

- 2b)

Geeignet, da nur jeder Knoten des Zertifikats durchgegangen und seine Nachbarn überprüft werden müssen und hier sogar dieser naive Ansatz mit Aufwand O(|E| * |V|), welcher im worst case ($|E| = |V|^2$) zu O($|V|^3$) wird, bewältigbar ist.

- 2c)

Nicht geeignet, da auch wenn es eine Wahrheitsbelegung gibt die Phi nicht erfüllt trotzdem eine andere existieren könnte die Phi erfüllt.

- 2d)

Geeignet, da die Breitensuche in Polynominalzeit durchführbar ist.

- 2e)

Nicht geeignet, da eine andere Clique >k existieren könnte, die jedoch keine Obermenge von U ist.

- 3a)

P kann auf Q in O(n^3 * log n) übersetzt werden. P <=p Q.

Q kann in O(m^2) gelöst werden.

Daher kann P in O((n^3 * log n)²) = O(n^6 * (log n)²) gelöst werden.

- 3b)

P <=p Q

Wahr weil alle Probleme in NP auf NP-vollständige reduziert werden können. Kann nicht beantwortet werden. Wahr wenn Q auf P reduziert werden könnte, falsch wenn nicht. Wahr, weil Q auf SAT reduziert werden kann und somit auch P auf SAT reduziert werden kann. Kann nicht beantwortet werden. Wahr wenn die Reduktion P auf SAT (über beliebig viele Zwischenschritte z.B. Q) in O(n) möglich ist, falsch wenn nicht.

- 4a)

Zertifikat: Ein Guardian Set X von G mit |X| <= k Zertifizierer: Überprüfe, ob jeder Knoten in V X zu mindestens einem Knoten aus X adjazent ist.

- 4b)

k=k

U = V

Für jeden Knoten v e V erzeugen wir eine Menge S_v = {u e V: v adjazent zu u}, also all seinen Nachbarn und sich selbst.

S ist die Menge aller S_v.

Reduktion ist klarerweise polynomiell.

Korrektheit:

Sei X = {v1, ..., vk} ein Guardian Set von G. Dann ist X'= {S_v_1, ... S_v_k} ein Set Cover von S da per Definition alle Knoten V X zu X mindestens einem Knoten in X adjazent sind und somit in Summe alle Knoten des Graphen in den Teilmengen enthalten sind.

Sei X' = {S_v_1, ... S_v_k} ein Set Cover von S. Dann ist X = {v1, ... v_k} ein Guardian Set von G da wenn man alle S_v_x vereinigt alle Knoten enthalten sind und in S_v_x alle Nachbarn von v_x enthalten sind. Somit sind alle Knoten in X mit allen anderen V X benachbart.