

# Beschränkte Suchbäume

Finde Verzweigungsregeln für diverse verschiedene Probleme, die zeigen, dass das jeweilige Problem in FPT ist. Wie gut die Laufzeit genau ist, ist dann erstmal nicht so wichtig.

## Cluster Editing

---

### Problemstellung

Gegeben einen Graphen  $G$  und einen Parameter  $k$ , kann man  $G$  mit bis zu  $k$  Operationen in einen Cluster-Graphen überführen? Dabei ist eine Operation das Einfügen oder Löschen einer Kante.

Ein Cluster-Graph ist eine disjunkte Vereinigung von Cliques.

### Lösung

Wir haben genau dann eine disjunkte Vereinigung von Cliques, wenn es keinen induzierten  $P_3$  gibt.

Seien nun  $u, v, w$  in induzierter  $P_3$ . Dann müssen wir mindestens eine der drei Operationen machen: die Kante  $\{u, w\}$  einfügen, die Kante  $\{u, v\}$  löschen oder die Kante  $\{v, w\}$  löschen. Damit haben wir eine Verzweigung der Weite 3 und in jedem Zweig wird der Parameter um eins kleiner.

## Independent Set

---

### Problemstellung

Gegeben ein  $d$ -degenerierten Graph  $G$  und ein Parameter  $k + d$ , gibt es in  $G$  ein Independent Set der Größe  $k$ ?

Ein Graph ist  $d$ -degeneriert, wenn jeder Subgraph einen Knoten mit Grad höchstens  $d$  hat.

### Lösung

Wähle einen Knoten  $v$  mit Grad höchstens  $d$ . Wir verzweigen auf der Entscheidung einen der Nachbarn von  $v$  zu wählen, wofür es bis zu  $d$  Optionen gib. Zusätzlich gibt es die Option, dass wir keinen der Nachbarn wählen. In dem Fall ist es aber nie falsch  $v$  selbst zu wählen. Damit erhalten wir bis zu  $d+1$  Zweige und in jedem wird ein Knoten ausgewählt, was  $k$  um eins verkleinert. Für die neue Instanz in jedem Teilbaum löschen wir den gewählten Knoten und alle seine Nachbarn.

## Closest String

---

## Problemstellung

Gegeben  $k$  Strings und einen Parameter  $d$ , gibt es einen String dessen Hamming-Abstand zu jedem der  $k$  Strings maximal  $d$  ist?

## Lösung

Seien  $s_1, \dots, s_k$  die  $k$  gegebene Strings. Starte mit  $s = s_1$ . Wir können nun bis zu  $d$  Modifikationen an  $s$  vornehmen, um von den anderen  $s_i$  ebenfalls Abstand maximal  $d$  zu bekommen. Falls der Hamming-Abstand zu einem der anderen  $s_i$  größer als  $2d$  ist, dann können wir abbrechen und **nein** ausgeben (es gibt dann keinen String, der zu  $s$  und  $s_i$  Abstand  $d$  hat). Falls  $s$  zu allen  $s_i$  Abstand maximal  $d$  hat können wir ebenfalls abbrechen und **ja** ausgeben.

Andernfalls, sei  $s_i$  ein String mit Abstand  $\delta$  vom aktuellen  $s$  mit  $d < \delta \leq 2d$ . Dann müssen wir für eine gültige Lösung mindestens eine der  $\delta$  Stellen von dem aktuellen  $s$  ändern. Damit haben wir eine Verzweigung mit Weite  $\leq 2d$ , bei der in jedem Zweig die Anzahl noch verfügbaren Änderungen um eins kleiner wird.

Beachte, dass die Anzahl verfügbarer Änderungen dann ggf. nicht mehr gleich  $d$  ist. Konkret, wenn  $x < d$  die Anzahl verfügbarer Änderungen ist, dann bleibt der zu erreichende Zielabstand weiterhin  $d$ . Das heißt, wir wissen, dass  $d < \delta \leq x + d$  und eben nicht  $x < \delta \leq 2x$ .

## Odd Cycle Transversal

---

### Problemstellung

Gegeben einen perfekten Graph und einen Parameter  $k$ , gibt es eine Knotenmenge der Größe  $k$ , die nicht-leeren Schnitt mit jedem ungeraden Kreis in hat?

Ein Graph ist perfekt, wenn für jeden induzierten Teilgraph die Cliquenzahl  $\omega$  gleich der chromatischen Zahl  $\chi$  ist.

### Lösung

Jeden ungeraden Kreis zu treffen bedeutet, dass der Graph nach Löschen der gewählten Knoten bipartit ist. Wenn der aktuelle Graph bereits bipartit ist, dann sind wir fertig. Falls nicht, dann ist die chromatische Zahl und damit auch die Cliquenzahl größer 2. Damit gibt es insbesondere ein Dreieck und von diesem Dreieck müssen wir mindestens einen Knoten auswählen. Wir haben also eine Verzweigung der Weite 3 und der Parameter wird in jedem Schritt eins kleiner.

## Feedback Vertex Set in Tournaments

---

## Problemstellung

Gegeben einen Turniergraphen und einen Parameter  $k$ , kann man  $k$  Knoten löschen, sodass der resultierende Graph azyklisch ist?

Ein Turniergraph ist ein gerichteter Graph, bei dem jedes Knotenpaar durch genau eine Kante verbunden ist.

## Lösung

Für den Fall, dass der Graph  $G$  noch nicht azyklisch ist, zeigen wir, dass es ein gerichtetes Dreieck enthält. Sei  $C$  ein möglichst kurzer gerichteter Kreis. Wenn  $C$  kein Dreieck ist, dann gibt zwei in  $C$  nicht benachbarte Knoten  $u, v$ . Da  $G$  ein Turniergraph ist, gibt es die Kante  $uv$  oder  $vu$ ; ohne Einschränkung  $uv$ . Diese Kante  $uv$ , zusammen mit dem Pfad von  $v$  nach  $u$  auf  $C$  liefert einen kürzeren gerichteten Kreis. Damit gibt es ein gerichtetes Dreieck.

Von diesem Dreieck muss mindestens ein Knoten gelöscht werden. Das liefert eine Verzweigung mit Weite 3 und der Parameter wird in jedem Schritt eins kleiner. Beachte, dass die resultierenden Teilgraphen wieder Turniergraphen sind.

## Feedback Arc Set in Tournaments

---

### Problemstellung

Gegeben einen Turniergraphen und einen Parameter  $k$ , kann man  $k$  Kanten löschen, sodass der resultierende Graph azyklisch ist?

Ein Turniergraph ist ein gerichteter Graph, bei dem jedes Knotenpaar durch genau eine Kante verbunden ist.

### Lösung

Man kann hier die selbe Idee wie bei [Feedback Vertex Set in Tournaments](#) versuchen. Das Problem dabei ist, dass man nach dem Löschen von Kanten keinen Turniergraphen mehr hat. Jetzt gibt es zwei Möglichkeiten. Entweder man argumentiert, dass man weiterhin einen kurzen gerichtete Kreis findet (Länge von  $k$  abhängig) oder wir umgehen das Problem mit folgender Aussage.

**Lemma:** Es gibt genau dann ein Feedback Arc Set der Größe maximal  $k$ , wenn es eine Kantenmenge  $A$  mit  $|A| \leq k$  gibt, sodass das Umdrehen jeder Kante aus  $A$  den Graphen azyklisch macht.

Wenn wir nach einem möglichst kleinen Feedback Arc Set suchen, dann können wir also Kanten umdrehen, statt Kanten zu löschen. Das restliche Argument geht analog zu [Feedback Vertex Set in Tournaments](#).

Um das Lemma zu beweisen, beachte dass die Richtung von *“Umdrehen macht azyklisch”* zu *“Löschen macht azyklisch”* offensichtlich ist. Betrachte also nun den Fall, dass es ein Feedback Arc Set der Größe maximal  $k$  gibt und sei  $A$  ein minimales Feedback Arc Set. Wir zeigen, dass das Umdrehen von  $A$  den Graph azyklisch macht. Sei  $G'$  der Graph, den man durch Umdrehen der Kanten in  $A$  erhält und nimm an, dass  $G'$  einen Kreis enthält. Sei  $C$  ein solcher Kreis mit minimaler Anzahl umgedrehter Kanten (beachte, dass  $C$  mindestens eine umgedrehte Kante enthält, da  $C$  sonst auch ein Kreis in  $G - A$  wäre). Sei  $uv \in A$  eine Kante, sodass  $vu$  auf  $C$  liegt. Da  $A$  ein minimales Feedback Arc Set ist muss  $G - A + uv$  zyklisch sein. Es gibt also in  $G - A$  einen gerichteten Pfad von  $v$  nach  $u$ . Damit können wir  $vu$  in  $C$  durch diesen Pfad ersetzen, der keine umgedrehte Kante enthält und bekommen einen Kreis der weniger umgedrehte Kanten enthält als  $C$ ; ein Widerspruch.